



*Ariel
Hardware Reference
Library*

Technical Reference



*Ariel
Hardware Reference
Library*

Technical Reference

IBM Internal Use Only

Ariel Technical Reference

Table Of Contents

GENERAL

Ariel Systems Overview

PROCESSOR

Hazel-A Hardware Functional Specification

COMMUNICATIONS

Attached Processor Source Communications Adapter (APSCA)

ADAPTERS

TPAC-A Hardware Functional Specification

EPROM4-A Hardware Functional Specification

Attached Processor Communication Sub-System Functional Specification

INPUT/OUTPUT

Ariel User Manual (I/O Cards)

Ariel I/O Unit Test System Manual

32 Bit DI/DO-A Hardware Functional Specification

Ariel System Analog to Digital Converter Functional Specification

Ariel System Digital to Analog Converter Functional Specification

MEMORY

Ariel System Memory Card Functional Specification

APPENDIX

Distributed Sensor Subsystem:Diagnostic/Utility User's Manual

ARIEL Systems Overview

Control Systems Engineering

**IBM Corporation
General Technology Division
D-035 B-300-47A
East Fishkill
tel 533-7654**

IBM Internal Use Only

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

PREFACE

This document is an overview of the ARIEL Distributed Sensor Subsystem (ARIEL DSS) hardware. It deals with the ARIEL system as a whole and does not go into detail on any specific card. If the reader requires this type of information, it can be found in the cards' functional specifications, the titles of which are listed below:

1. ARIEL HAZEL-A Functional Specification
2. ARIEL Memory Functional Specification
3. ARIEL Remote Distributed Interface System Functional Specification
4. ARIEL Attached Processor/Communications Sub-System Functional Specification
5. ARIEL Attached Processor/Source Communications Adapter Functional Specification
6. ARIEL Hazel Input/Output Processor Functional Specification

This document is intended as a guide to users wishing to configure a tool controller using the ARIEL hardware. It specifies the basic function of each ARIEL card and gives sample configurations. It also provides timings and pin definitions for the ARIEL Bus (ABUS) for those users wishing to develop hardware for the ARIEL system.

It is assumed that the reader is familiar with the Distributed Sensor Subsystem (DSS) architecture from which ARIEL was derived. If the reader is not familiar with DSS, background information can be obtained in the following documents:

1. Distributed Interface System, System Summary, Form No. Z45-0918.
2. Distributed Sensor Sub-system User Guide, Vol. 1, Form No. Z45-1123.
3. Distributed Sensor Sub-system User Guide, Vol. 2, Form No. Z45-1124.

Any questions concerning the information in this document should be directed to:

GTD - East Fishkill Facility
Dept/035 Bld/300 Zip/47A
Rt. 52 Hopewell Jct. NY 12533
teline 533-7654

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

TABLE OF CONTENTS

Introduction	1
Packaging	1
Function Cards	1
Mother Board	2
Logic Block	3
Architecture	5
Engine	5
System Control Store	5
Attached Processors	5
AP/SCA	5
AP/CSS	6
Macrofunction	6
Mother Board	6
ARIEL DSS Configurations	7
ARIEL DSC	7
ARIEL RDIS	10
ARIEL BUS	11
Pin Definitions	11
Control Store Interface	11
Attached Processor Interface	14
Macrofunction Interface	14
Miscellaneous ABUS lines	15
Timing Diagrams	16

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

LIST OF ILLUSTRATIONS

Figure 1.	System Board Component Diagram	2
Figure 2.	ARIEL Logic Block Package	4
Figure 3.	ARIEL DSS Logic Block Structure	6
Figure 4.	ARIEL Mother Board Structure	7
Figure 5.	ARIEL DSC-IV Configuration	8
Figure 6.	ARIEL DSC-V Configuration	9
Figure 7.	ABUS Pin Assignments (J1-J8).	12
Figure 8.	ABUS Pin Assignments (JJ1)	13
Figure 9.	Request/Acknowledge Timing	16
Figure 10.	Cycle Steal Read Timing	16
Figure 11.	Cycle Steal Write Timing	17
Figure 12.	Control Store Read Timing (Engine)	18
Figure 13.	Control Store Write Timing (Engine)	18
Figure 14.	Macrofunction Interface Read/Write	21
Figure 15.	ARIEL Bus I/O Operations	22

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

INTRODUCTION

ARIEL is a flexible general-purpose industrial controller for plant floor automation. It consists of a set of function cards which interface to a common bus to provide general-purpose processing, data storage, communications, and sensor-based I/O capabilities. These function cards are incorporated in a modular package to provide the user with maximum flexibility when configuring a system. ARIEL is designed to meet the needs of a diversified user base, function is added only as required by the application.

The ARIEL hardware is derived from the Distributed Sensor Subsystem IV (DSS-IV) hardware and incorporates many of the same functions; however, some changes have been made to increase the systems usability and to accommodate for packaging differences between the two systems. The user should consult the functional specifications of the individual cards for specific changes. The major changes are listed below:

1. Program memory has been doubled to 8 partitions of RAM and 1 partition of EPROM (each partition contains 32k words). The number of address lines has been increased to allow access to a total of 33 partitions.
2. An adapter has been designed for ARIEL to connect to the IBM ISLANDS Local Area Network which uses the IEEE standard 802.5 token ring protocol.
3. The remote I/O communications protocol has been modified to increase the performance; however, an adapter card is available for ARIEL which allows hook-up to existing remote DIS hardware.
4. The native disk support has been dropped. This function is expected to be replaced by disk servers on the local area network.
5. The parallel multidrop I/O channel has been dropped. ARIEL allows a number of I/O cards to be plugged into the same board as the controller, additional cards if required for the application communicate to the controller over a high-speed serial link.

The ARIEL package is smaller than the DSS-IV's, more flexible, and more reliable due to a reduction in the number of components. It has a common bus so that all card configurations are supported without the use of individually wire-wrapped boards. And it has no internal cables or power harnesses.

PACKAGING

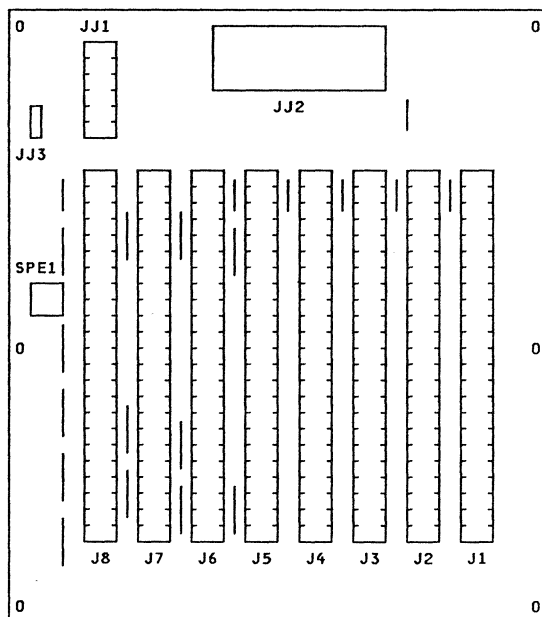
The ARIEL Distributed Sensor Subsystem (DSS) controller uses a modular packaging technique which groups logic into blocks with independent power supplies and enclosures. These logic blocks communicate to each other through a high-speed serial channel and can be configured as intelligent controllers, I/O concentrators, or a combination of the two with the use of Function Cards. This packaging technique allows the ARIEL DSS to change in size and function with the application requirements.

FUNCTION CARDS

Function cards are the first level of packaging in the ARIEL DSS. These cards have approximate dimensions of 11.5" X 4" and house IBM and vendor technology logic to provide the system with the following functions: general-purpose processing, data storage, communications support, connections to industrial I/O. The user selects from a family of these cards to customize the ARIEL package to the application.

All Function Cards interface to a common bus through a 120-contact edge connector located at the bottom of each card. This common bus, called the ARIEL BUS (ABUS), provides power to the cards and a means of trans-

IBM Internal Use Only



COMPONENTS: J1-J8 120-contact ABUS connectors
 JJ1 20-contact ISA/LSI connector
 JJ2 36-pin power connector
 JJ3 +24 vdc fan connector
 SPE1 system oscillator

Figure 1. System Board Component Diagram

ferring data between them. See "ARIEL Bus" for more information on the ABUS.

MOTHER BOARD

The ARIEL Mother Board illustrated in Figure 1 is designed to distribute the ABUS to a maximum of 8 of the above mentioned Function Cards. It is comprised of:

- Eight 120-contact female connectors (J1-J8) which mate with the edge connectors on the Function Cards. These connectors distribute the ABUS.
- A 20-contact female connector (JJ1). This connector is an extension to the ABUS which is available only in slot 8 to support the Block Interface function.

IBM Internal Use Only

- A 36-pin input power connector (JJ2) for direct hook-up to the ARIEL power supply.
- A 4-pin output power connector (JJ3) for the fan.
- Pull-up resistors for open collector drivers on the ABUS.
- System oscillator (SPE1).

Function cards can plug into any one of the eight slots on the Mother Board. The only condition is that the Function Card performing the Block Interface (BIC) function be located in slot 8.

LOGIC BLOCK

The ARIEL Logic Block is the second level of packaging in the ARIEL system after the Function Cards. It is comprised of Mother Board, a power supply, an enclosure, and up to 8 Function Cards as illustrated in Figure 2. The Mother Board plugs directly into the power supply with a 36-pin connector and distributes the 7 DC voltage levels to the ABUS connectors through internal power planes. The Function Cards plug into the ABUS connectors on the Mother Board and are held in place by mounting brackets that lock in place to form the front panel shown in Figure 2. From the front panel, the user has access to the I/O points on the Function Cards.

IBM Internal Use Only

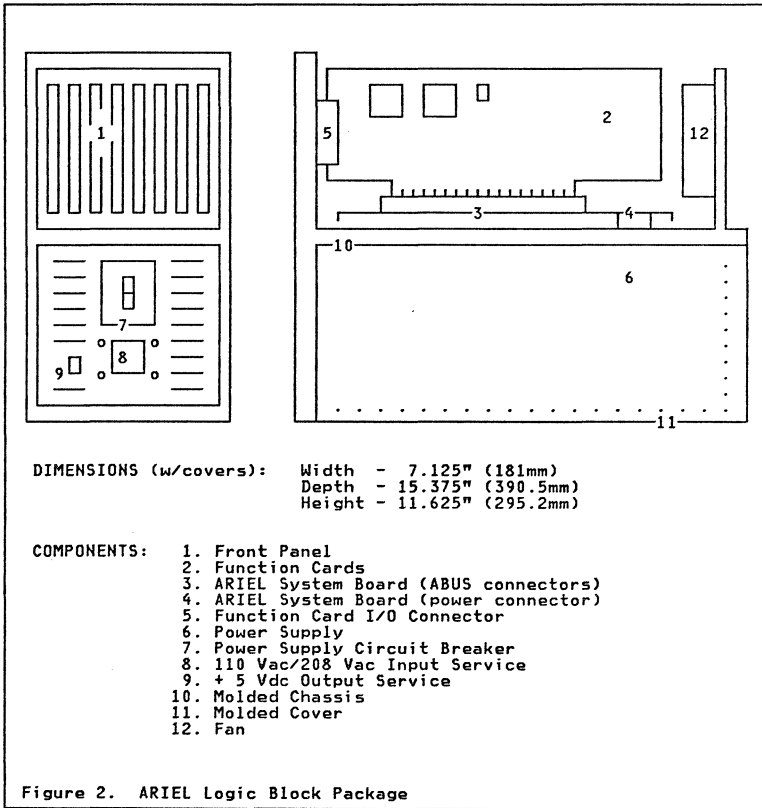


Figure 2. ARIEL Logic Block Package

IBM Internal Use Only

ARCHITECTURE

The block diagram in Figure 3 is an illustration of the physical structure of an ARIEL DSS Logic Block showing how the system bus (ABUS) connects the four different types of Function Cards - Engine, System Control Store, Attached Processor, Macrofunction - which comprise the ARIEL DSS architecture. The Engine (ENG) card provides general-purpose processing capability within the Logic Block. The System Memory card provides data and program storage capability for the ENG. The Attached Processor (AP) cards act as slaves to the ENG providing it with intelligent interfaces to other computers, consoles, or other Logic Blocks. And the Macrofunction cards provide the means of attaching to a tool's sensor-based I/O points.

ENGINE

One ENG card is allowed per Logic Block for general purpose processing. When installed in a Logic Block, this card takes control of the Control Store Interface (CSI) which it uses to fetch instructions and data from the System Memory card. In this release of ARIEL, the ENG card function is supported by the HAZEL-A card which is designed around the register-oriented JIB' microcontroller. The HAZEL-A has the following features:

- 16-bit JIB' Microcontroller
- AMD-9511 APU
- 500 ns. Cycle Time
- 288K Word memory addressing capability
- 256X18 Local Storage Registers
- 8 prioritized interrupt levels
- 7 prioritized cycle steal request lines
- Time-of-day counter
- Hardware interval timer
- Interrupt level switch in 2 microsec.
- Parity maintained on all data busses and control store address bus
- 8 diagnostic LEDs
- 1 process check LED
- 4 user programmable DIP switches

SYSTEM CONTROL STORE

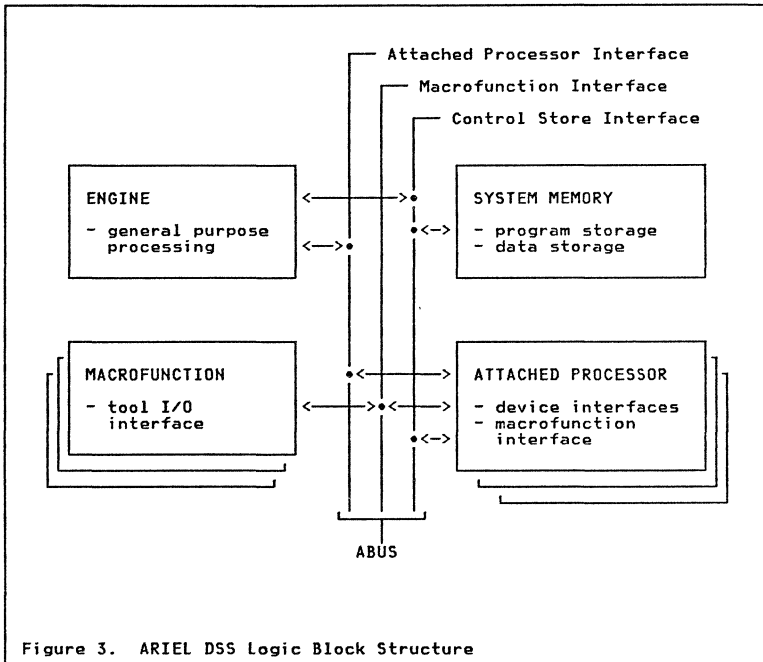
The System Memory card is organized as nine partitions of 32K words each. One partition is in EPROM, and eight are in RAM.

ATTACHED PROCESSORS

AP/SCA

The Attached Processor/Source Communications Adapter (AP/SCA) provides long distance communication to remote units over co-axial cable.

IBM Internal Use Only



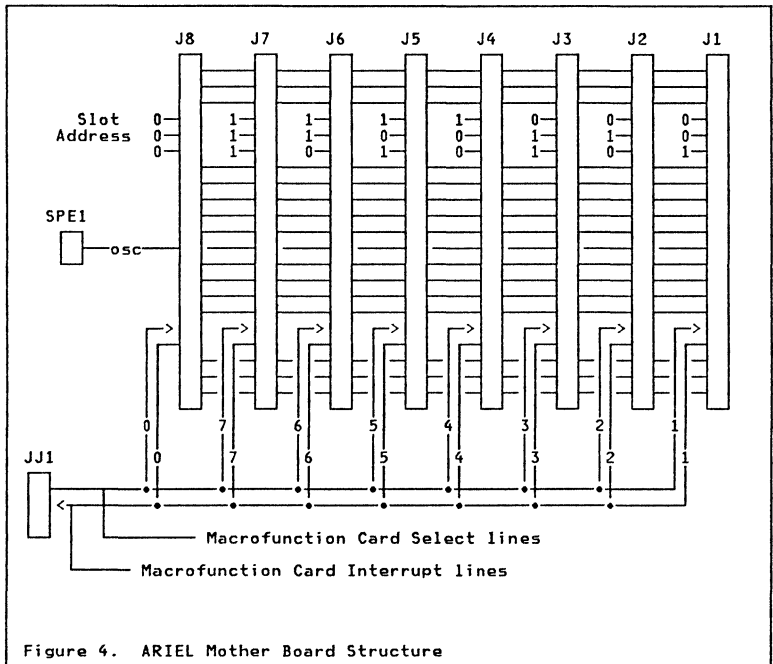
AP/CSS

The Attached Processor/Communications Sub System (AP/CSS) utilizes an optically isolated interface for medium distance, and an RS232-C port for short distance communications.

MACROFUNCTION

- Destination Communications Adapter (DCA) card which performs the Macrofunction Interface Controller (MFIC) function for the ARIEL RDIS.
- 32-bit Digital Input/Digital Output card.
- 16 channel Analog to Digital Converter card.
- 8 channel Digital to Analog Converter card.
- THORPE Programmable Adapter Card (TPAC).

MOTHER BOARD



ARIEL DSS CONFIGURATIONS

ARIEL DSC

The Distributed System Controller (DSC) is a general purpose controller with interfaces to a host computer, a system console, and an I/O channel. The host computer interface provides bidirectional communications to a computer of higher intelligence allowing the DSC to connect to a network of computers in an automation system with distributed intelligence. The system console interface provides a communication path to the operator. And the I/O channel interface provides attachment to a sub-network of lower intelligence I/O concentrators.

Figure 5 shows a Logic Block configured as a DSC-IV. This configuration provides a compatible replacement to existing DSS-IV hardware for users who do not require local disk support. The ARIEL DSC-IV is comprised of:

1. An ENG card for general processing of control algorithms.
2. A System Memory card for program/data storage.
3. An AP/CSS card with an RS-232C interface for short-medium distance communications (e.g. console) and an optically isolated interface for medium-long distance communications (e.g. host).

IBM Internal Use Only

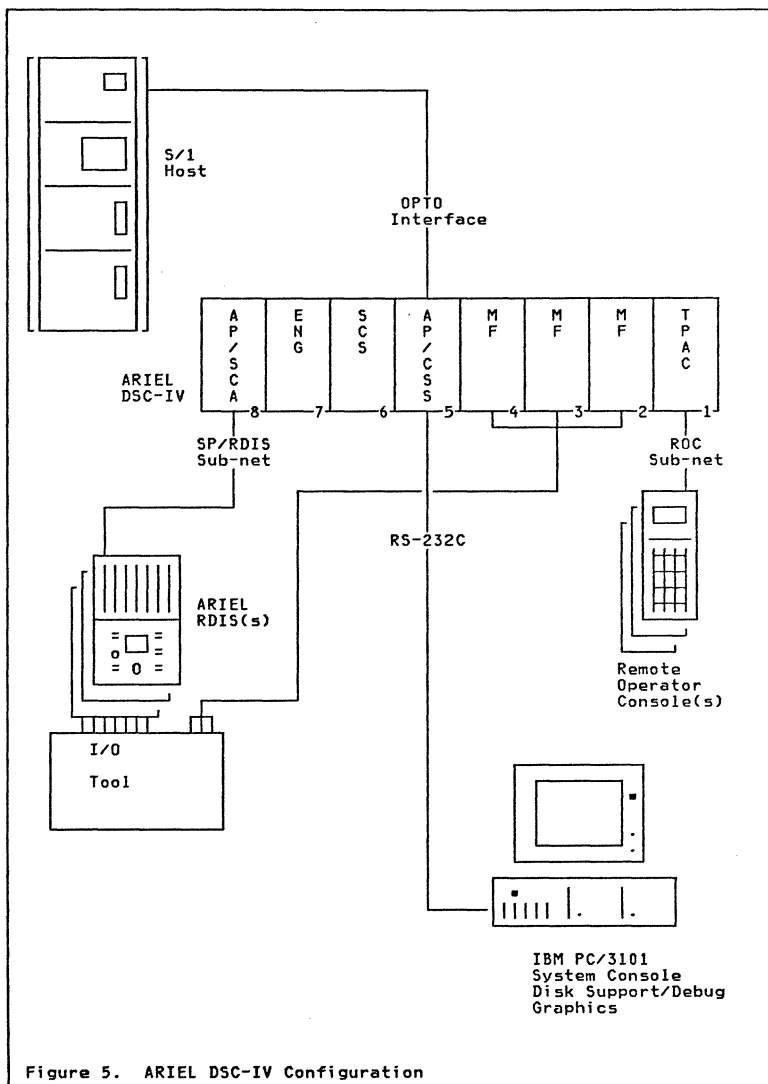


Figure 5. ARIEL DSC-IV Configuration

4. An AP/SCA card for connection to the Standard Protocol (SP/RDIS Sub-Network).
5. A TPAC card in the auto-poller mode for connection to the Remote Operator Console (ROC) sub-network.

IBM Internal Use Only

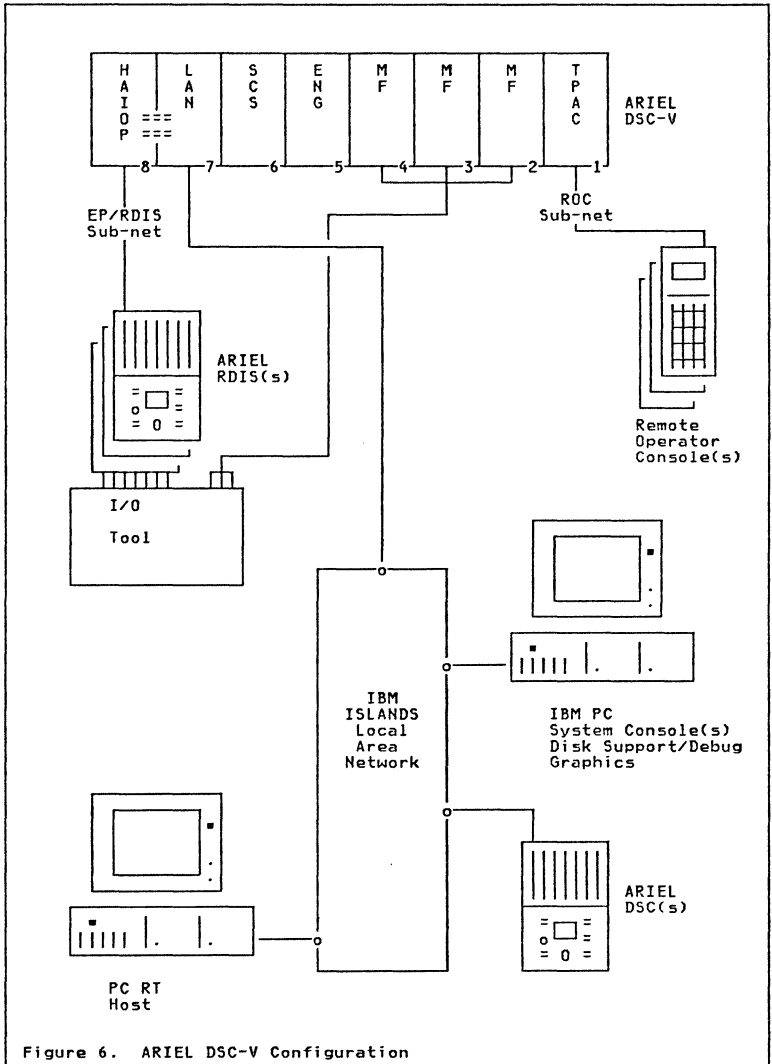


Figure 6. ARIEL DSC-V Configuration

Figure 6 shows a Logic Block configured as a DSC-V. This configuration utilizes the ENG, SCS, and TPAC cards in the same way as the ARIEL DSC-IV but replaces the AP/SCA and AP/CSS cards with the HAIOP and LAN cards. The HAIOP card provides a connection to the Enhanced Protocol (EP)/RDIS Sub-Network. It also attaches to the LAN card through a top card cross-over connector to provide the ARIEL DSC-V with a connection to the IBM

IBM Internal Use Only

ISLANDS Local Area Network on which are implemented the host communication, system console and peer-to-peer communication functions.

In both configurations, the remaining slots on the Mother Board can be used for MF cards. The placement of the cards in the Logic Block does not have to be as shown with the exception of the AP/SCA and HAIOP cards which must be in slot 8 to perform the Block Interface (BIC) function if MF cards are installed.

ARIEL RDIS

The ARIEL RDIS is a Logic Block configured as an I/O concentrator box. This configuration consists of up to 7 MF cards in slots 1-7 and the DCA or HAIOP card in slot 8. The DCA card is used if the ARIEL RDIS is to attach to the SP/RDIS Sub-Network which is compatible to that used on existing DSS hardware. The HAIOP is used if the ARIEL RDIS is to attach to the EP/RDIS Sub-Network which allows for variable length data block transfers.

IBM Internal Use Only

ARIEL BUS

The ARIEL Bus (ABUS) is comprised of the following 3 interfaces: 1) Control Store Interface (CSI), 2) Attached Processor Interface (API), and 3) Macrofunction Interface (MFI).

PIN DEFINITIONS

Control Store Interface

The CSI provides the Engine (ENG) and Attached Processors (APs) with read/write access capability to System Control Store (SCS). The ENG uses this interface to fetch instructions and data from the System Memory card. APs use this interface to transfer data between their on-board memory and SCS.

Control Store Data (-CS D0-15) This is the 16-bit bidirectional data bus between the processor cards and system memory, with bit 0 as the most significant bit.

Control Store Data Parity (-CS DP1-2) These two lines DP1 and DP2 are the odd parity bits for CS D0-7 and CS D8-15 respectively.

Control Store Partition Selects (-CS PA0-5) These 6 address lines select memory partitions 1 through 32 in the SCS for a 33 partition system. If only 9 partitions of memory are implemented, PA0-2 are used to select partitions 1 through 8 and PA3-5 are not used. In both cases, PA0 is the most significant bit.

Partition Zero Select (+P0 Select) This line selects memory partition 0 in the SCS and is independent of PA0-5. When active, partition 0 of control store is selected regardless of the value of the other partition select lines.

Control Store Address (-CS A1-15) These 15 address lines define an address location within the 32K word partition boundary in the SCS.

Control Store Address Parity (-CS Ap) This line carries the odd parity for the control store address lines including the 7 partition select lines.

Control Store Select (-CS Select) This control line selects the SCS card for a read/write operation. This line should only go active after the data and address busses are stable.

Control Store Write (-CS Write) This control line distinguishes between an SCS read or write operation. When low, data is written to the SCS. When high, data is read from the SCS.

Control Store ECC Error (-CS ECC Error) This line is brought low by the SCS card when it detects an ECC error during a memory read cycle. This line is active for both correctable (single bit) and uncorrectable (two or more bits) errors.

Control Store Uncorrectable Error (-CS Unc Error) This line is brought low by the SCS card when it detects an uncorrectable (two or more bits) ECC error during a memory read cycle.

Control Store Data Parity Error (-CS Dp Error) This line is brought low by the SCS card during a memory write operation if odd parity is not detected across each byte of the data bus. This error condition inhibits the completion of the write operation on the System Memory card.

IBM Internal Use Only

+ 15v	A01	—	•A0	00•	—	A02	AGND
- C/S Req 6	A03	—	•A1	01•	—	A04	- CS DP2
- C/S Req 5	A05	—	•A2	02•	—	A06	- CS DP1
- C/S Req 4	A07	—	•A3	03•	—	A08	- CS D15
- C/S Req 3	A09	—	•A4	04•	—	A10	- CS D14
- C/S Req 2	A11	—	•A5	05•	—	A12	- CS D13
- C/S Req 1	A13	—	•A6	06•	—	A14	- CS D12
- C/S Req 0	A15	—	•A7	07•	—	A16	- CS D11
- C/S Ack 6	A17	—	•A8	08•	—	A18	- CS D10
- C/S Ack 5	A19	—	•A9	09•	—	A20	- CS D9
- C/S Ack 4	A21	—	•B0	10•	—	A22	- CS D8
- C/S Ack 3	A23	—	•B1	11•	—	A24	- CS D7
- C/S Ack 2	A25	—	•B2	12•	—	A26	- CS D6
- C/S Ack 1	A27	—	•B3	13•	—	A28	- CS D5
- C/S Ack 0	A29	—	•B4	14•	—	A30	- CS D4
+ C/S RD Str	A31	—	•B5	15•	—	A32	- CS D3
+ 8.5v	A33	—	•B6	16•	—	A34	- CS D2
- AP Trigger	A35	—	•B7	17•	—	A36	- CS D1
- Slot Addr 2	A37	—	•B8	18•	—	A38	- CS D0
- Slot Addr 1	A39	—	•B9	19•	—	A40	- CS ECC Error
- Slot Addr 0	A41	—	•C0	20•	T	A42	- CS Select
- Int Req 4	A43	—	•C1	21•	—	A44	- CS Unc Error
- Int Req 3	A45	—	•C2	22•	T	A46	- CS Write
- Int Req 2	A47	—	•C3	23•	—	A48	+ 12v
- CS Any Error	A49	—	•C4	24•	—	A50	- CS Dp Error
- Int Req 0	A51	—	•C5	25•	—	A52	- CS Ap Error
- Run Error	A53	—	•C6	26•	T	A54	- CS A15
(spare)	A55	—	•C7	27•	T	A56	- CS A14
(spare)	A57	—	•C8	28•	T	A58	- CS A13
- CS PA5	A59	T	•C9	29•	T	A60	- CS A12
- CS PA4	A61	T	•D0	30•	T	A62	- CS A11
- CS PA3	A63	T	•D1	31•	T	A64	- CS A10
- Key Bit	A65	—	•D2	32•	T	A66	- CS A9
- Key Bit Check	A67	—	•D3	33•	T	A68	- CS A8
(spare)	A69	—	•D4	34•	T	A70	- CS A7
- 5v	A71	—	•D5	35•	T	A72	- CS A6
- POR	A73	—	•D6	36•	T	A74	- CS A5
- RESET	A75	—	•D7	37•	T	A76	- CS A4
+ 24 mhz osc	A77	—	•D8	38•	T	A78	- CS A3
+ Inh DP Check	A79	—	•D9	39•	T	A80	- CS A2
- P0 WR Enable	A81	T	•E0	40•	T	A82	- CS A1
- P0 WR Error	A83	—	•E1	41•	T	A84	+ P0 Select
- EPE	A85	T	•E2	42•	T	A86	- CS PA2
- EES	A87	T	•E3	43•	T	A88	- CS PA1
- 15v	A89	—	•E4	44•	T	A90	- CS PA0
- Diag Sel	A91	T	•E5	45•	T	A92	- CS Ap
- I/O Int Req	A93	—	•E6	46•	—	A94	- 12v
- I/O SYNC OUT	A95	—	•E7	47•	—	A96	- I/O D7
- I/O RET IN	A97	—	•E8	48•	—	A98	- I/O D6
- I/O Intrpt	A99	—	•E9	49•	—	B00	- I/O D5
- I/O Select	B01	—	•F0	50•	—	B02	- I/O D4
- I/O C3	B03	—	•F1	51•	—	B04	- I/O D3
- I/O C2	B05	—	•F2	52•	—	B06	- I/O D2
- I/O C1	B07	—	•F3	53•	—	B08	- I/O D1
- I/O C0	B09	—	•F4	54•	—	B10	- I/O D0
- I/O SYNC IN	B11	—	•F5	55•	—	B12	- I/O Dp IN
- I/O RET OUT	B13	—	•F6	56•	—	B14	- I/O Dp OUT
+ 5v	B15	—	•F7	57•	—	B16	- GND
+ 5v	B17	—	•F8	58•	—	B18	- GND
+ 5v	B19	—	•F9	59•	—	B20	- GND

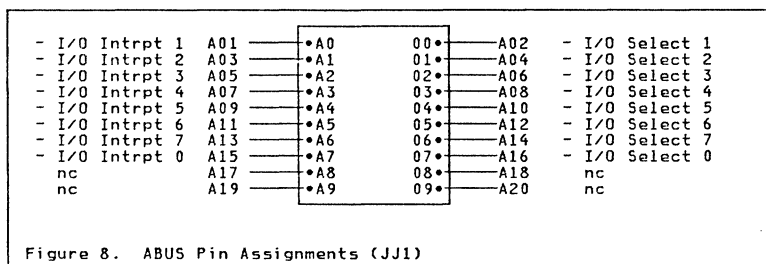
0=Open Collector

T=Tri-state

Figure 7. ABUS Pin Assignments (J1-J8).

Control Store Address Parity Error (-CS Ap Error) This line is brought low by the System Memory card during a read or write operation if it does not detect odd parity across the address bus. This

IBM Internal Use Only



error condition inhibits the completion of the write operation on the System Memory card but does not affect the read operation.

Partition Zero Write Error (-P0 WR Error) This line is brought low by the System Memory card if a protected area of partition 0 is written to. This error condition inhibits the completion of the write operation on the System Memory card. The range of the protected area is hardware programmable on the System Memory card.

Control Store Any Error (-CS Any Error) This line is brought low by the System Memory card if any of the following error conditions occur: ECC uncorrectable error, address parity error, data parity error, or partition zero write error.

Partition Zero Write Enable (-P0 WR Enable) When active, this line enables the processor cards to write to protected areas of partition 0 by disabling the partition 0 write protect logic on the System Memory card.

EEPROM Program Enable (-EPE) When electrically erasable PROMs are implemented on the System Memory card, this line in conjunction with CS Write are held low by the processor card to enable programming. If UV erasable PROMs are implemented on the System Memory card, this line is not used.

Extended EEPROM Select (-EES) In a 9 partition system (e.g. only CS PA0-PA2 are used), partition 8 is reserved for EEPROM/EPROM and partitions 0-7 for RAM. If more than 1 partition of EEPROM/EPROM is implemented on the System Memory card, this line allows the EEPROM/EPROM address to overlap the RAM address. In this system, a processor card brings this line low to access an EEPROM/EPROM address which overlaps a RAM address.

Control Store Key Bit Input (-Key Bit) When implemented on the System Memory card, the Key Bit input is used as a 17th data bit in the generation of the 6-bit ECC stored with each word in memory. The ENG card holds this line low when it grants a cycle steal operation to an AP and the RESET line is not active. When the RESET line is active or the ENG is accessing the System Memory, this line is held high.

Control Store Key Bit Check (-Key Bit Check) The System Memory card brings this line low during a memory read cycle if the Key Bit input value does not match the value of this input when that location was written. For example, when the ENG writes to a location in memory, the Key Bit input used to generate the ECC is high. When the AP reads this same location, the Key Bit input used to check the ECC is low causing a check condition. The same occurs if an AP writes to a location and the ENG reads that location.

IBM Internal Use Only

Attached Processor Interface

The API is comprised of a number of lines which facilitate communication between the ENG and the APs.

Cycle Steal Request (-C/S Req 0-6) These are 7 prioritized cycle steal request lines to the ENG, C/S Req 0 being the highest priority. A low level on any one of these lines will be detected by the ENG and recognized as a request to free the CSI for one cycle. C/S Req 0 is used as the refresh request line from the System Memory card. The other 6 are available to APs.

Cycle Steal Acknowledge (-C/S Ack 0-6) These are the acknowledge lines for the above cycle steal request lines. When the ENG detects a cycle steal request, it sets the appropriate acknowledge line low for one cycle and tri-states its CSI drivers to give the AP access to the SCS.

Cycle Steal Read Strobe (+C/S RD Str) This line is high when data from the System Memory card is valid. AP cards use the falling edge of this line to latch data during a cycle steal read operation.

Interrupt Request (-Int Req 0,2-4) These are 4 prioritized interrupt request lines to the ENG, Int Req 0 being the highest priority. When low, the processor on the ENG card will be interrupted and forced to execute on the appropriate level provided the interrupt has not been masked off. The interrupt for level 1 is not implemented on the ABUS. This interrupt has a dedicated function on the ENG card.

Attached Processor Trigger (-AP Trigger) This line is brought low by the ENG to interrupt APs. The AP to be interrupted is determined by the value of the three least significant bits of the control store address (CS A13-A14) which are compared by the AP to the slot address lines (Slot Addr 0-2) for the slot in which it resides.

Engine Reset (-RESET) When this line is held low by an AP, the processor on the ENG card is reset to address zero.

Macrofunction Interface

The MFI consists of an 8-bit data bus and various control lines for handshaking. It provides an interface between the Block Interface Card (BIC) located in slot 8 of the Mother Board and the MF cards in slots 1-7.

I/O Data (-I/O D0-D7) This is an 8-bit bidirectional data bus between the BIC and the MF cards, bit 0 being the most significant bit.

I/O Data Parity Out (-I/O Dp OUT) This line is driven by the BIC and carries the odd parity for the data bus during an I/O write operation.

I/O Data Parity In (-I/O Dp IN) This line is driven by the MF card and carries the odd parity for the data bus during an I/O read operation.

I/O Command (-I/O C0-C3) This is a 4-bit command bus from the BIC to the MF cards.

I/O Sync Out (-I/O SYNC OUT) This control line is brought low by the BIC after it sets up the data and command busses. A low level on this line in conjunction with a low level on the I/O Select line forces the MF card to decode the I/O command bus lines and execute the command.

I/O Return In (-I/O RET IN) This control line is brought low by the MF card in response to the I/O SYNC OUT line going active. A low level on this line is interpreted by the BIC as meaning that

IBM Internal Use Only

the MF card has completed the command issued to it and that the data bus contains valid data (for a read operation).

I/O Sync In (-I/O SYNC IN) This control line is brought low by an MF card after it sets up the data and command busses. A low level on this line forces the BIC to put a low level on the I/O SYNC OUT line. This line gives a MF card the capability of talking to another MF through the BIC.

I/O Return Out (-I/O RET OUT) This control line is brought low by the BIC when the I/O RET IN goes active or when its own return line goes active.

I/O Select (-I/O Select) When low, this line enables the MF card to respond to the I/O SYNC OUT line. This line is not common to all slots on the ABUS. Each slot has an individual I/O select line which is driven by the BIC through a special 20-pin connector on the Mother Board. The BIC selects a MF card by activating the appropriate I/O Select line.

I/O Interrupt (I/O Intrpt) This line allows a MF card to cause a process interrupt on the BIC which, in turn, relays it to the ENG card. The interrupt condition occurs if the value of this line does not match the value of its corresponding reference bit stored in a reference register on the BIC. This line is not common to all slots on the ABUS. Each slot has an individual interrupt line which is input to the BIC through a special 20-pin connector on the Mother Board.

I/O Interrupt Request (-I/O Int Req) This line is brought low by the BIC if there is a process interrupt pending condition on the card, and the interrupt enable switch in its control register has been set.

Miscellaneous ABUS lines

The following control lines do not fall within the three major categories mentioned above and are described here individually.

Diagnostic Select (-DIAG Select) This line can be used by an AP as an alternate to the CS Select line during a cycle steal operation. It provides the AP with cycle steal capability to a card other than the System Memory card.

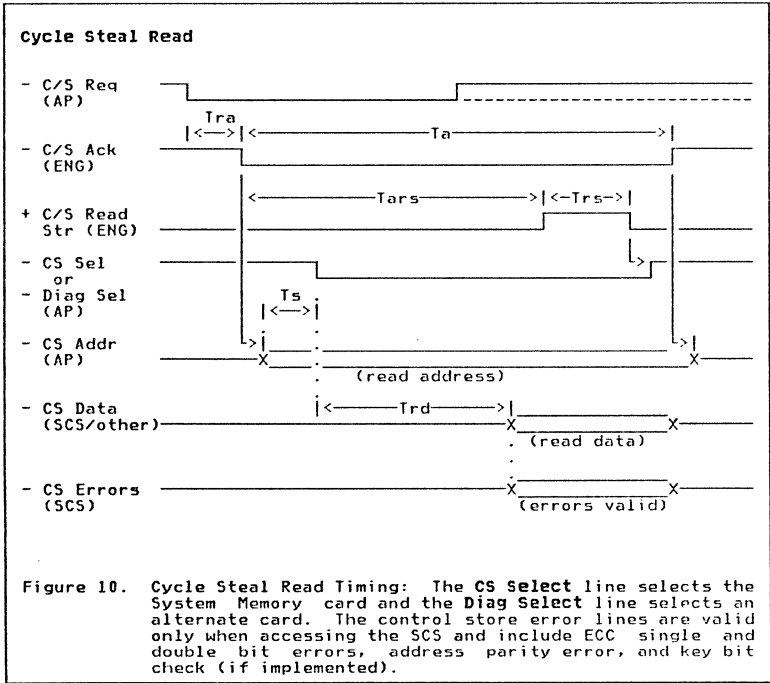
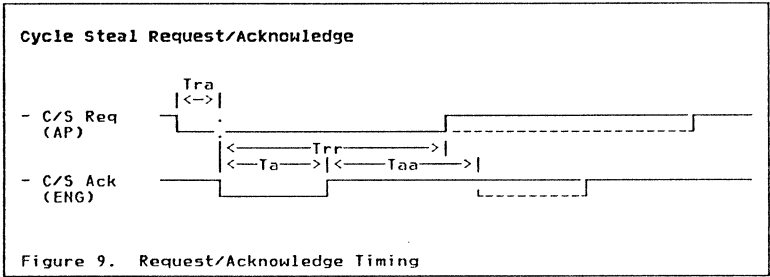
Power-On-Reset (-POR) This line is held low for a short period of time during initial power-up to allow all cards in the system to reset to a known state before beginning execution.

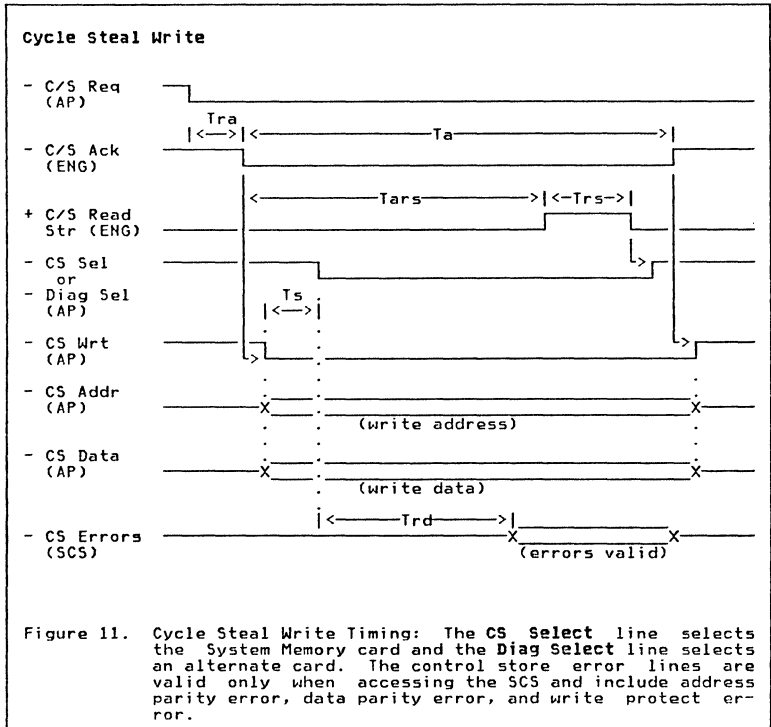
Processor Run Error (-Run Error) This line is brought low by any one of the processor cards in the system as an indication that it has failed and cannot continue processing. Other processor cards can interrogate this line to determine the status of the system.

System Oscillator (+24 MHZ OSC) This is a 24 megahertz square wave signal used for timing control on the various cards in the system.

Slot Address Lines (-Slot Addr 0-2) These 3 lines define the address of each slot on the 8 slot ARIEL Mother Board, bit 0 being the most significant bit.

TIMING DIAGRAMS





IBM Internal Use Only

Control Store Read (Engine)

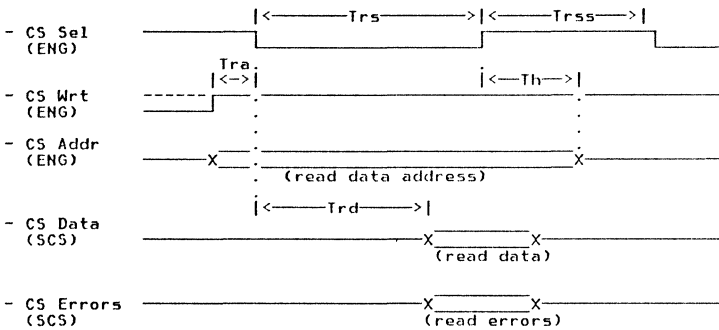


Figure 12. Control Store Read Timing (Engine): The Control Store read errors include ECC single and double bit errors, address parity error, and key bit check (if implemented).

Control Store Write (Engine)

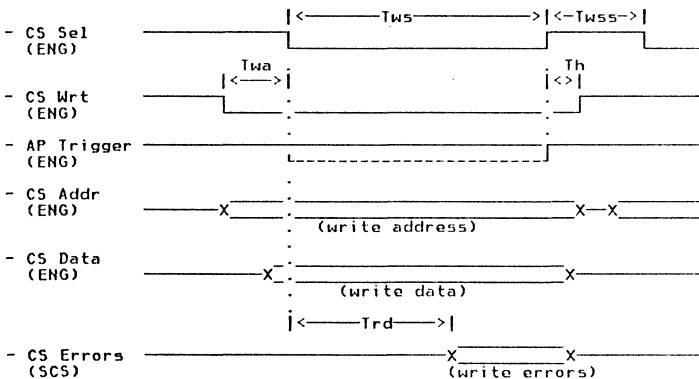


Figure 13. Control Store Write Timing (Engine): The AP TRIGGER signal follows the CS SELECT line when the ENG writes to locations X'0010' thru X'0017' of partition 0. The Control Store write errors include address parity error, data parity error, and write protect error.

IBM Internal Use Only

CYCLE STEAL INTERFACE TIMING PARAMETERS

Symbol	Description	MIN ns	ACT ns	MAX ns	see note
Tra	ACK active from REQ	50		550	1,3
Ta	ACK pulse width		417		1
Trr	single request reset time	0		917	1,4
Tars	READ STRB active from ACK		292		1,5
Trs	READ STRB pulse width		83		1,5
Ts	address/data setup time before CS SEL	10		40	
Trd	data/error valid from CS SEL			250	2
Taa	ACK to ACK (continuous request)		583		1,4

The above table applies to Figure 9, Figure 10, and Figure 11.

NOTE 1: These timings apply when the HAZEL-A card is used as the ENG.

NOTE 2: These timings apply when the ARIEL 1/2 MB System Memory card with DOVER RAM modules is used as the SCS.

NOTE 3: The ACK line is guaranteed to go active within the time limits specified provided there are no other higher priority requests active.

NOTE 4: If the REQ line is not reset within the time limit specified, the HAZEL-A will recognize it as a new cycle steal request and respond with an acknowledge.

NOTE 5: READ STRB is generated by the HAZEL-A near the end of the cycle. APs use the falling edge of this signal to reset the select line and to latch data during a read operation.

IBM Internal Use Only

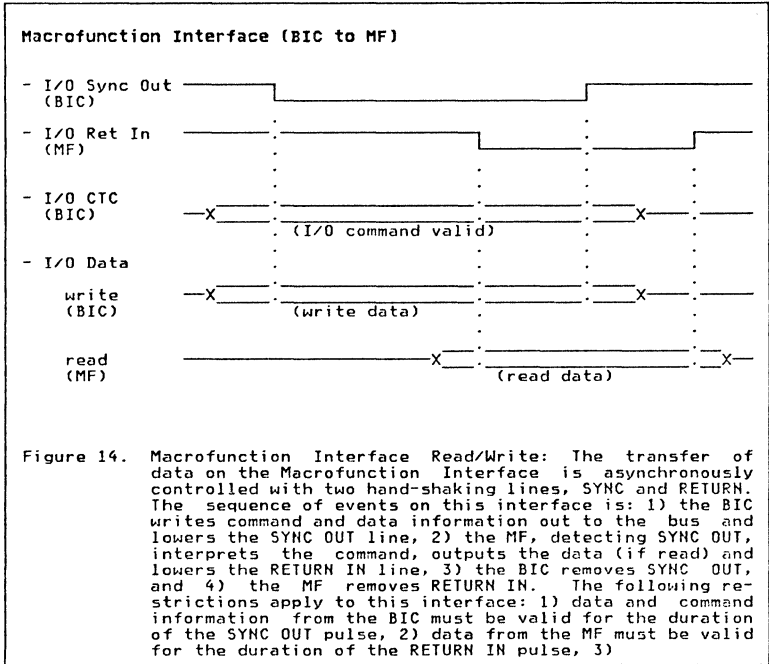
ENGINE/CONTROL STORE INTERFACE TIMING PARAMETERS

Symbol	Description	MIN ns	ACT ns	MAX ns	see note
Trs	read SEL pulse width		292		1
Trss	SEL off to SEL on following a read	185			1.6
Tra	address setup before SEL for a read		41		1
Tws	write SEL pulse width		333		1
Twss	SEL off to SEL on following a write	100			1.6
Twa	address setup before SEL for a write		83		1
Th	address hold after SEL	80			1
Trd	data/error valid from SEL			250	2

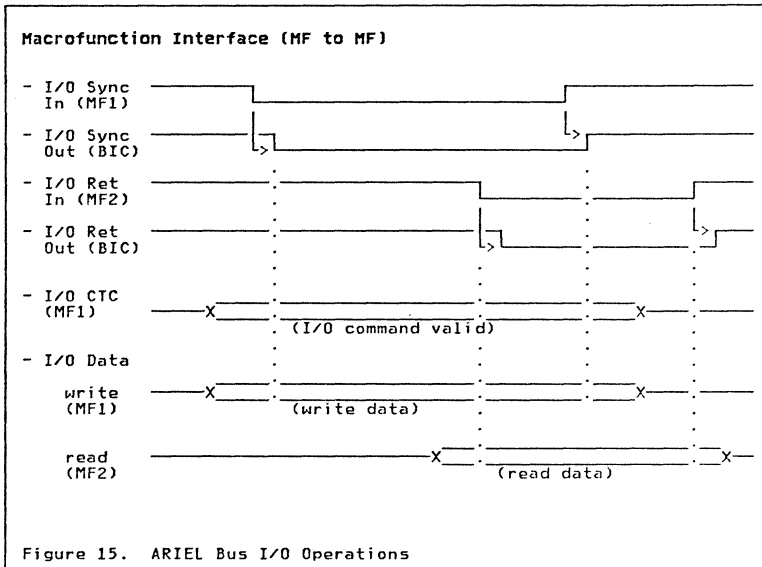
The above table applies to Figure 12, and Figure 13.

NOTE 6: The minimum time occurs when the ENG grants a cycle steal acknowledge immediately following a control store read/write cycle.

IBM Internal Use Only



IBM Internal Use Only



Heading ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Heading References</u>
abus	ABUS	11	ARIEL Bus 2

Figure ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Figure References</u>
planar	PACK	2	1: 2
unit	PACK	4	2: 3, 3
sysblk	ARCH	6	3: 5
sysb	ARCH	7	4: 7
config4	ARCH	8	5: 7
config5	ARCH	9	6: 9
abus	ABUS	12	7: 19
abus2	ABUS	13	8: 19
cst1	ABUS	16	9: 19
cst2	ABUS	16	10: 19
cst3	ABUS	17	11: 19
rdtime	ABUS	18	12: 20
wrttime	ABUS	18	13: 20
iot1	ABUS	21	14: 15
iot2	ABUS	22	15: 15

Imbed Trace

Page ii	PREFACE
Page viii	INTRO
Page 1	PACK
Page 3	ARCH
Page 10	ABUS

HAZEL-A HARDWARE FUNCTIONAL SPECIFICATION

Control Systems Engineering

**IBM Corporation
General Technology Division
D-035 B-300-47A
East Fishkill
tel 533-7654**

IBM Internal Use Only

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

TABLE OF CONTENTS

Hazel-A Functional Specification	1
General Information	1
Conventions	1
Reference Documentation	1
Part Number Information	1
Features	1
Card Overview	2
Hazel-A Data Flow	3
A-BUS Interface Definition	5
A-BUS Inputs and Outputs	6
User Interface	10
Led's	10
Hardware Switches	11
Memory Interface	11
Addressing	12
Timing	14
Errors	15
Cycle Steal Interface	16
Level Definition	16
AP Trigger	18
Jib Prime Processor	18
Arithmetic Processor Unit	18
Data Formats -- Command/Data	18
Interface to JIB PRIME	21
APU Round-off Errors	21
Clock Control	21
Hazel-A Interrupt Theory	24
Level Definition	24
LSR Addresses for PSW (Int Level 0-7)	24
Interrupt Handler	25
Level 0 Interrupt (machine check)	26
Timing	27
Hazel-A Timers and Frequency Dividers	27
Funnel	29
Local Storage Registers TOMCAT	30
Timing	30
Addressing	30
External Registers	32
Timing	33
Definitions and Addressing	33
Hazel-IV to Hazel-A Comparison	40
Appendix A. Test Connectors	41
Appendix B. APU Programming Notes	43

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

LIST OF ILLUSTRATIONS

Figure 1.	Block diagram of the Hazel-A card	2
Figure 2.	HAZEL-A DATA FLOW	3
Figure 3.	HAZEL-A ABUS 2nd LEVEL	5
Figure 4.	Memory Organization	11
Figure 5.	MEMORY ADDRESSING SCHEME	13
Figure 6.	Control Store Fetch Timing	14
Figure 7.	Control Store Write Timing	15
Figure 8.	CYCLE STEAL INTERFACE TIMINGS	17
Figure 9.	Command Format, APU Interface	19
Figure 10.	Status Format, APU Interface	19
Figure 11.	Data Format, Fixed Precision	20
Figure 12.	Data Format, Floating Point Precision	20
Figure 13.	CLOCK MODULE 2nd LEVEL DIAGRAM	23
Figure 14.	INTERRUPT HANDLER 2nd LEVEL DIAGRAM	25
Figure 15.	Interrupt Timing (PSW SWAP)	27
Figure 16.	INTERVAL, 1 SEC TIMER MODULE 2nd LEVEL DIAGRAM	28
Figure 17.	FREQUENCY DIVIDER MODULE 2nd LEVEL DIAGRAM	29
Figure 18.	FUNNEL 2nd LEVEL DIAGRAM	30
Figure 19.	LSR Addressing Scheme	31
Figure 20.	EXTERNAL REGISTER 2nd LEVEL DIAGRAM	32
Figure 21.	GROUP 2 -- External Register Module (U26)	34
Figure 22.	GROUP 3 -- Interval/1 sec Timer Module (U21)	35
Figure 23.	GROUP 4 -- Frequency Divider Module (U23)	36
Figure 24.	GROUP 5 -- External Register Module (U25)	37
Figure 25.	GROUP 6 -- Clock Module (U20)	38
Figure 26.	GROUP 7 -- Interrupt Module (U22), JIB' Module (U20)	39
Figure 27.	TEST CONNECTOR (JJ1)	41
Figure 28.	TEST CONNECTOR (JJ2)	41
Figure 29.	TEST CONNECTOR (JJ3)	42

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

HAZEL-A FUNCTIONAL SPECIFICATION

GENERAL INFORMATION

This card is used as the Engine in the Ariel System. It is assumed the user is familiar with the JIB' functional specification.

Conventions

1. Logical conventions are defined where used
2. Hex numbers are represented in negative logic
+ 5v --> logical 0
gnd --> logical 1

A signal's active state is defined by the + or - preceeding the signal name. ie. '- PO WR EN' is active when that line is ground or a logical 1.

3. Bit 0 -> Most Significant Bit
4. Even -> more significant than odd

Reference Documentation

JIB ' Functional Specification

EDX Manual

All IBM modules are released and have Release Interface Tapes (RIT),
logics are handled through Dept 035, East Fishkill.

Part Number Information

Assembly --> 6912775
Raw Card --> 6912776
Schematic --> 6912777

EC --> A37493J

FEATURES

1. JIB' Microprocessor
2. Supports EDX-J
3. Supports PASCAL
4. On Board APU
5. 1 Interval Timer
6. 1 One Second Timer
7. 8 Interrupt Levels
 - 5 Hardware
 - 3 Software
8. Interrupt Level Swap in 666.67 nsec max once acknowledged
9. Enhanced Int Level 0, Monitoring Memory Errors and JIB' Errors
10. 19 Memory Address Lines with Parity; Address up to 288K words
11. Can Address up to 32 External Registers
12. Can Address up to 256 x 18 Local Storage Registers
13. Memory timing controlled through Clock Control Module
 - Cycle Time = 333.33 nsec without memory access
 - Cycle Time = 500.0 nsec with memory access
14. Designed for A-BUS Compatability
15. Easy Access to Test Points
16. Run Error Indicator
17. 8 LEDs Available for Diagnostics
18. 3 Hardware Programmable Switches Available to Software through XR's

CARD OVERVIEW

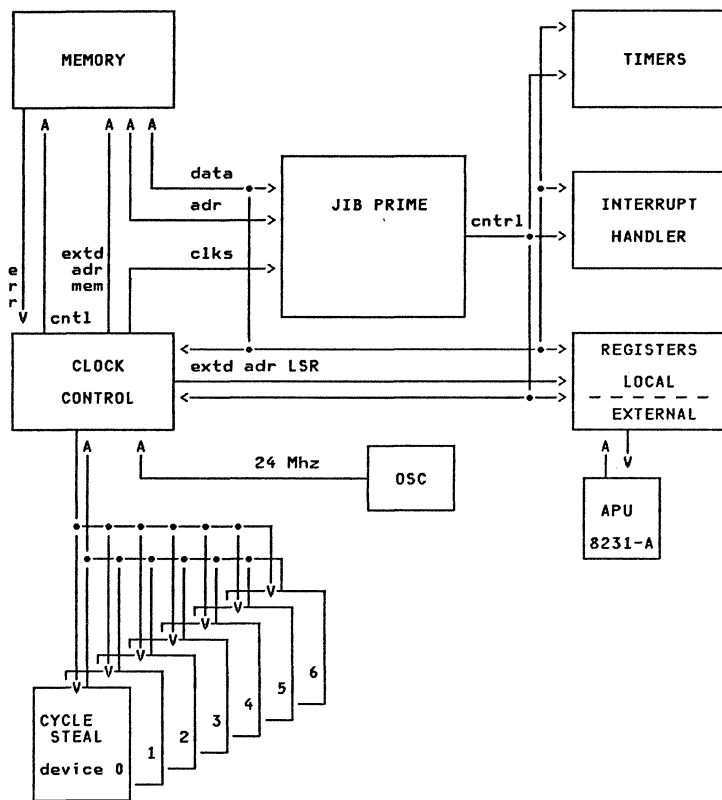


Figure 1. Block diagram of the Hazel-A card: The memory and attached devices are separate cards.

Hazel-A interfaces to memory, which is off card, through JIB's two 18 bit busses (DB0 and DBI), 19 address bits, (16 from JIB' and 3 extended bits from the clock module), and Control and error handling which also is handled through the clock module and miscellaneous vtl logic. JIB' communicates to all external registers with the DB0 and DBI side of it's data bus. These busses and the DB0 and DBI EVEN make up the LSR data bus. Addressing to local registers is accomplished through the LSR address bus from JIB' and 3 bits of extended address from the clock module. Addressing to external registers is accomplished through the XR address bus from JIB'.

IBM Internal Use Only

Also on card are an Arithmetic Processor Unit (APU), a one second timer and an interval timer. These can interrupt the JIB¹ and are accessible through the external register bus. The Interrupt handler module shown, provides 8 levels of interrupt and is also accessible through the external register bus.

Hazel-A allows attached devices access to Hazel-A memory through a cycle steal interface. This interface which entails clock control is handled by the clock module. Not shown in Figure 1 on page 2 is the funnel module. It allows LSR data to/from memory, and also multiplexes memory data, LSR data, or XR data to JIB's DBI bus. See Figure 2 for more details.

HAZEL-A DATA FLOW

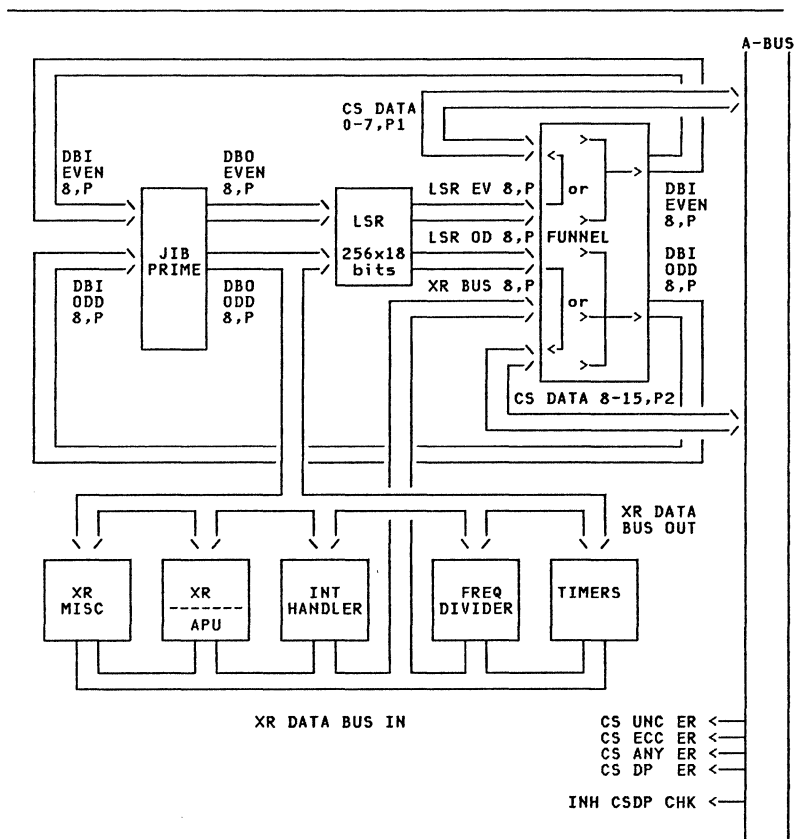


Figure 2. HAZEL-A DATA FLOW

IBM Internal Use Only

JIB' has an 18 bit Data Bus IN (DBI) and an 18 bit Data Bus Out (DBO). Since the Data bus is not bidirectional, timing control and data flow become very simple. The only way to input data to JIB' is through the DBI Bus. Thus LSR data, Control Store data, or XR data (ODD side only) is readable through the funnel to JIB'. The only way to output data from JIB' is through the DBO Bus. Thus LSR's, Control Store, or External Registers (ODD side only) are written to via the DBO Bus.

Parity is generated and checked through out Hazel-A. The DBI bus has a parity checker/generator in JIB' which causes an internal error when bad parity is detected. This can be inhibited for devices without parity with the Inhibit Control Store Data Parity Check line. When this signal is active Parity will not be checked but will be generated for incoming DATA only, NOT INSTRUCTIONS. Thus Hazel-A cannot execute out of non-parity memory. Also, Parity is maintained on the DBO Bus. Thus all External Registers must and do generate Parity but do not check parity.

The following section will define all A-BUS signals used by Hazel-A. This is to familiarize the user with the I/O Hazel-A uses on A-BUS and terminology that will be used further on in this specification.

IBM Internal Use Only

A-BUS INTERFACE DEFINITION

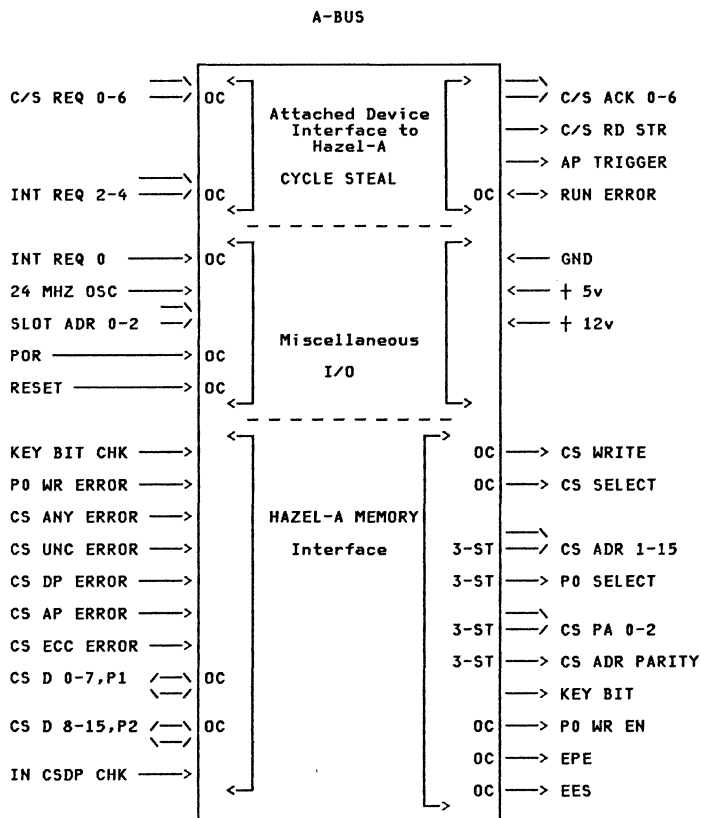


Figure 3. HAZEL-A ABUS 2nd LEVEL: oc - open collector, 3-st - tri-state, all the rest are vtl push-pull

IBM Internal Use Only

A-BUS Inputs and Outputs

The Ariel Bus is a common bus between all cards in the system. This includes any processor cards (Attached processor, Hazel-A or Memory), and any macro-function cards (32 bit DI/DO, T-PAC, CAP, A-D, DAC). The bus consists of a memory cycle steal bus designed for the Hazel-A and also The DIS bus for I/O capability.

The following is a list of all the I/O on the Hazel-A and their definitions as a function of the Hazel-A card. These I/O are shown in Figure 3 on page 5.

- C/S REQ (0-6) I Cycle Steal Request (0-6); 7 Prioritized inputs (0 being the highest priority) used via Attached Devices to access Hazel-A memory. Hazel-A will give up at most 50% of it's cycles to Cycle Stealing Devices.

- C/S ACK (0-6) O Cycle Steal Acknowledge (0-6); 7 outputs corresponding respectively to the 7 Cycle Steal Request signals. A grant, from Hazel-A to the Attached Device, that Hazel-A memory is available to that device.

Any one of the 7 acknowledges being active, tri-states Hazel-A from memory.

+ C/S RD ST O Cycle Steal Read Strobe ; This output acknowledges that data from Hazel-A memory is valid during a Hazel-A or Cycle Steal read.

- AP TRIG O Attached Processor Trigger ; This output is used via Hazel-A to interrupt any Attached Device. This signal will be active when Hazel-A writes to memory locations between address X'0010' to X'0017'.

- X'0010' -> Interrupts Attached Device in slot 8
- X'0011' -> Interrupts Attached Device in slot 1
- X'0012' -> Interrupts Attached Device in slot 2
- X'0013' -> Interrupts Attached Device in slot 3
- X'0014' -> Interrupts Attached Device in slot 4
- X'0015' -> Interrupts Attached Device in slot 5
- X'0016' -> Interrupts Attached Device in slot 6
- X'0017' -> Interrupts Attached Device in slot 7

Note: This signal causes P0 WR EN to go active

- SLOT ADR (0-2) I Slot Address ; These three inputs are hardwired on the A-bus mother board to a predefined value between B'000' and B'111'. They will be used via Hazel-A to determined which slot on the board it is located.

- Slot 1 --> B'001'
- Slot 2 --> B'010'
- Slot 3 --> B'011'
- Slot 4 --> B'100'
- Slot 5 --> B'101'
- Slot 6 --> B'110'
- Slot 7 --> B'111'
- Slot 8 --> B'000'

Note: logical 0 = + 5v and logical 1 = gnd

These signals are available through XR '09' bits 0-2.

- INT REQ (2-4) I Interrupt Request (2-4); 3 pulse triggered interrupts which are sampled at the end of the cycle (T7). The minimum required pulse width is 25 nsec. These signals are also maskable through the interrupt handler module. See "Hazel-A Interrupt Theory" on page 24 for more information.

IBM Internal Use Only

These signals are available through XR '1C', bits 2-4.

-CS ANY ER **I** Control Store Any Error ; This signal causes a level 0 interrupt to Hazel-A when memory access has failed. It is a combination of 4 possible memory access errors.

They are;

1. Control Store Uncorrectable Error
2. Control Store Data Parity Error
3. Control Store Address Parity Error
4. Partition 0 Write Error

This signal is not available through an XR bit, but the signals it comprises are.

- INT REQ 0 **I** Interrupt Request 0; A fourth, pulse triggered interrupt which is also sampled at T7 time. This signal is different than INT REQ 2-4 in that it is added with 3 other signals to generate the Level 0 Interrupt to the Interrupt Handler Module. Those signals are;

1. JIB ERROR OUT
2. CONTROL STORE ANY ERROR
3. KEY BIT CHECK

This signal is available through XR '1C' bit 0.

- RUN ERROR **I/O** Run Error Input ; This input to Hazel-A, from any device on the A-BUS, signals that that device is not running properly or that it has not finished bring-up (IPL) diagnostics. This signal is accessible via xr '09' bit 6.

Run Error Output ; Signals to all attached devices that the Hazel-A is not finished bring-up diagnostics (IPL) or that it is not running properly. This signal is accessible via xr '08' bit 7.

Note: This signal is good for a bring-up indicator but is very limited as a diagnostic tool. Any device can bring the signal active but no other device can tell which has gone down or is running improperly.

- KEY BIT **O** Key Bit ; Hazel-A output to the memory card. This signal is generated as follows,

- + 5v -> logical 0 -----> Hazel-A write to Hazel-A memory
- gnd -> logical 1 -----> Attached Device write to Hazel-A memory

This bit is stored in memory as part of 6 ECC bits.

Also this signal is generated from the cycle steal logic and is not an external register bit.

Note: No attached device should generate this signal.

- KEY B CHK **I** Key Bit Check ; Generated via the Memory card. When active this signal notifies Hazel-A (when reading memory) that the current Key Bit signal is different than the Key Bit signal when that location in memory was last written. This signal is maskable and is 1 of the level 0 interrupts. It is also available through XR '08' bit 6.

POR **I** Power On Reset ; This buffered input will cause Hazel-A to reset to a known state and when POR goes high Hazel-A starts execution at address X'0000' provided RESET is not active.

IBM Internal Use Only

RESET	I RESET ; When active, only JIB' on Hazel-A will reset to a known state and when RESET goes inactive the Hazel-A will begin execution at address X'0000' provided POR is not active.
+24 MHZ OSC	I 24 Mega-Hertz Oscillator ; The base clock frequency for Hazel-A. Also, used to generated the 3 MHZ clock for the APU.
+ IN CSDP CK	I Inhibit Control Store Data Parity Check ; When active, will inhibit parity checking over the DBI BUS on JIB', and will generate parity for the data on the DBI BUS. Any device Hazel-A can access through the Control Store data bus must guarantee parity with the data. If parity is not available, then this signal must be active when Hazel-A accesses the device. This signal will be or'd with xr '1E' hex (30 dec) bit 7, Inhibit DE/D0 parity check before entering the JIB'. Note: Since parity is checked on the DBI bus during instruction fetch, Hazel-A can not execute out of the EEPROMs (there is no parity on the EEPROM's).
- P0 WR EN	O Partition 0 Write Enable ; This output will allow writing into partition 0 of Hazel-A memory. Only that part of the 32K boundary defined on the memory card as being Protectable will be affected by this signal. This area will be called Protected Memory. See the Memory Card Functional Spec for the portion of partition 0 that will be affected. Thus if writing into the Protected Memory without having P0 WR EN active, a P0 WR Error will occur. This signal is available through XR '0B' bit 6. • gnd -> logical 1 -> Allows write to partition 0 of Hazel-A memory • + 5v -> logical 0 -> Disables write to Partition 0 of Hazel-A memory Note: AP trigger will cause this signal to go active.
- P0 WR ER	I Partition 0 Write Error ; While active, this signal causes a level 0 interrupt to Hazel-A. The active state is generated while writing to Protected memory without enabling P0 WR EN signal. This signal is available through XR '0B' bit 5.
- EPE	O Gated Eeprom Program Enable ; Enables programming of Partition 8 (EEPROM's). The output is available through XR '0B' bit 4.
- EES	O Extended Eeprom Select ; This signal is available through XR '0B' bit 5.
-CS DATA P2	I/O Control Store Data Parity 2 ; Bidirectional open-collector parity bit for control store data bits 8-15 (odd parity bit)
-CS DATA P1	I/O Control Store Data Parity 1 ; Bidirectional open-collector parity bit for control store data bits 0-7 (even parity bit)
-CS D(0-15)	I/O Control Store Data 0-15 ; Bidirectional open-collector data bus to/from memory. D0 is the most significant bit and D15 is the least significant bit.

IBM Internal Use Only

- CS D0 - D7 --> Control Store data even
- CS D8 - D15 --> Control Store data odd
- CS ECC ER I Control Store Ecc Error ; 16 of this type of an error will cause a Set/Reset latch to set. This will set XR '09' bit 7. This signal represents a Double bit error or single bit error while reading Hazel-A memory.
- CS SELECT O Control Store Select ; Output used to select Hazel-A memory. This signal is active during a read or write operation.
- CS UNC ER I Control Store Uncorrectable Error ; While active this signal causes a level 0 interrupt to Hazel-A. It represents a Double bit error while reading Hazel-A memory.

This signal is available through XR '08' bit 3.
- CS WRITE O Control Store Write ; Output used to determine direction of data to/from Hazel-A memory;

• high = +5v ---> Hazel-A read
• low = gnd ---> Hazel-A write
- CS DP ER I Control Store Data Parity Error ; While active this signal causes a level 0 interrupt to Hazel-A. It represents a Data parity error while writing to Hazel-A memory.

This signal is available through XR '08' bit 1.
- CS AP ER I Control Store Address Parity Error ; While active this signal causes a level 0 interrupt to Hazel-A. It represents an Address parity error while reading or writing Hazel-A memory. The address parity includes CS ADR 1-15, P0 SEL, and CS PA(0-2)

This signal is available through XR '08' bit 2.
- CS A(1-15) O Control Store Address (1-15) ; These Tri-state outputs along with P0 Select and CS PA0-2 address Hazel-A memory. These address lines select a specific address within a 32k boundary.
- +P0 Select O Partition 0 Select ; Tri-state output used with CS A(1-15) and CS PA(0-2) to address Hazel-A memory.

• Active ---> Allows Hazel-A to access Partition 0 (where the nucleus resides)

• Inactive --> Allows Hazel-A to access the Partition of memory selected by CS PA(0-2), instead of Partition 0.
- CS PA(0-2) O Control Store Page Address (0-2) ; These Tri-state outputs along with CS A(1-15) and P0 Select address Hazel-A memory.

IBM Internal Use Only

These 3 address lines select 1 of 8 partitions of Hazel-A memory excluding partition 0. Thus there are 9 partitions of 32k words (16 bits/word) in Hazel-A memory and are addressed as shown below;

CS PA0	CS PA1	CS PA2	P0 Sel	Partition Selected
X	X	X	0	0
1	1	1	1	1
1	1	0	1	2
1	0	1	1	3
1	0	0	1	4
0	1	1	1	5
0	1	0	1	6
0	0	1	1	7
0	0	0	1	8

Note: logical 0 = + 5v and logical 1 = gnd

-CS AP **0** Control Store Address Parity ; This tri-state output represents parity across CS A(1-15), P0 Select, and CS PA(0-2).

Voltages **I** There are 9 different voltage levels available on the Abus connector. They are;

Voltages		Pins	
1.	gnd	116, 118, 120	
2.	+ 5v	115, 117, 119	
3.	+ 12v	48	
4.	- 5v	71	not used
5.	+ 8.5v	33	not used
6.	- 12v	94	not used
7.	+ 15v	1	not used
8.	- 15v	89	not used
9.	Agnd	2	not used

Note: Agnd is Analog ground and should be kept separate from gnd (digital ground).

USER INTERFACE

Led's

There are 9 LED's available to the user. One is tied to the Hazel-A run error line. When on, the LED signals that the Hazel-A has crashed or is running irregularly. This LED is viewable outside the box enclosure the Hazel-A mounts in, and is available through XR '0B' hex bit 7. The remaining 8 LED's will mount on top of the card and will be available for diagnostics. These LED's are software defined and are available through XR '08' hex.

IBM Internal Use Only

Hardware Switches

On card are 4 dip switches. These switches are defined below.

1. Switch 1 ---> User defined Dip Switch. Available through XR '09' hex bit 3.
2. Switch 2 ---> User defined Dip Switch. Available through XR '09' hex bit 4.
3. Switch 3 ---> User defined Dip Switch. Available through XR '09' hex bit 5.
4. SWITCH 4 ---> Enables/Disables Level 1 Interrupts
 - ON Level 1 Interrupts are enabled
 - OFF Level 1 Interrupts are disabled

See "Appendix A. Test Connectors" on page 41 for Test point definitions and locations.

MEMORY INTERFACE

Hazel-A can address up to 288k words of memory, (Control Store, CS). This memory resides external to Hazel-A. Thus Hazel-A must interface to the memory card over the A-BUS. This interface entails the CS DATA BUS, CS ADDRESS BUS, and control and error signals to/from memory, all of which will be detailed in this section.

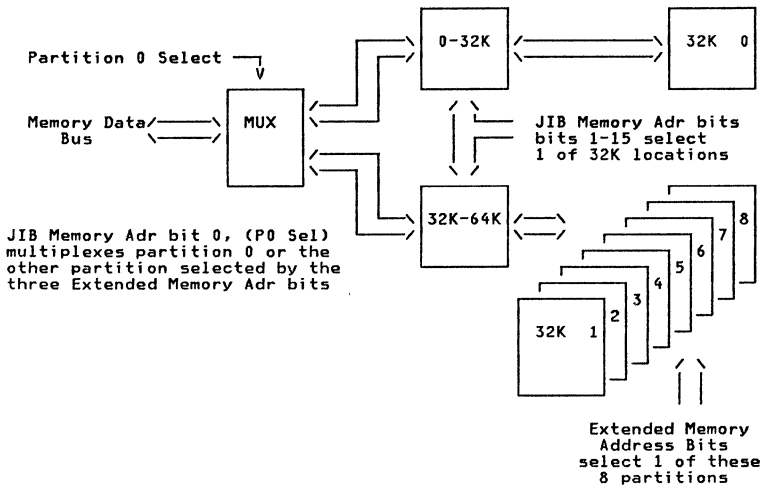


Figure 4. Memory Organization: Hazel-A has a 64K immediate address range. Thus it is always executing out of Partition 0 and one of the eight partitions defined by the 3 memory extended address bits.

Hazel-A can address up to 9 partitions (32k words each) of memory. JIB' can address up to 64K words of memory, without the Extended memory address

IBM Internal Use Only

bits. Thus in the case of the Ariel system, the firmware (nucleus) will reside in Partition 0 and application programs will reside in the remaining 8 partitions, giving access to the firmware at all times.

Addressing

Hazel-A addresses Control Store via 19 address lines. They are CS A(1-15), P0 Select, and CS PA(0-2). P0 select and CS A(1-15) are generated directly from JIB' as a 16 bit address, with bit 0 as the P0 Select line. The address range is expanded a factor of 8 by using the three CS PA (0-2) bits. These bits are implemented in the clock module.

The 15 CS Address lines address a single word within a 32k boundary, and the 4 other address lines are used to select 2 of 9, 32k word partitions that will make up the 64K immediate address range of Hazel-A. P0 Select, when active, allows access to Partition 0 and when inactive allows access to 1 of the 8 partitions addressed by CS PA(0-2). See Figure 4 on page 11 for a detailed explanation.

CS PA(0-2) signals are generated from the Instruction (I) register, or the Data (D) register or hardwired to a specific value and gated to these address lines dependent on P0 Select. See "A-BUS Inputs and Outputs" on page 6 for more details on CS PA (0-2) bits. The I and D registers will be explained further in the following paragraphs. Also, see "Clock Control" on page 21. If P0 select is active, the CS PA(0-2) lines are set high. If P0 select is inactive the CS PA(0-2) lines are generated from the I register or the D register, depending on the type of instruction being executed. If the micro-instruction is CS1 and bit 15 of the opcode is off (B'0') or the micro-instruction is CS0, the CS PA(0-2) lines are generated from the D register. If the micro-instruction is CS1 and bit 15 of the opcode is on (B'1'), the CS PA(0-2) lines are generated from the I register. See Figure 5 on page 13 for reference.

The 19 memory address bits to Hazel-A memory are gated through 3-st drivers with G Bit (Any cycle steal grant) and the Control signals are gated through OC drivers with G Bit. Thus if an Attached Device is granted a cycle, (G-Bit active), Hazel-A will be isolated from the memory bus. See "Cycle Steal Interface" on page 16 for more information on the Cycle Steal Interface.

Note: Hazel-A or any attached device on the A-bus accesses Control Store by words only. For byte operations on data in memory the word must be moved to LSR's and then the operation performed.

IBM Internal Use Only

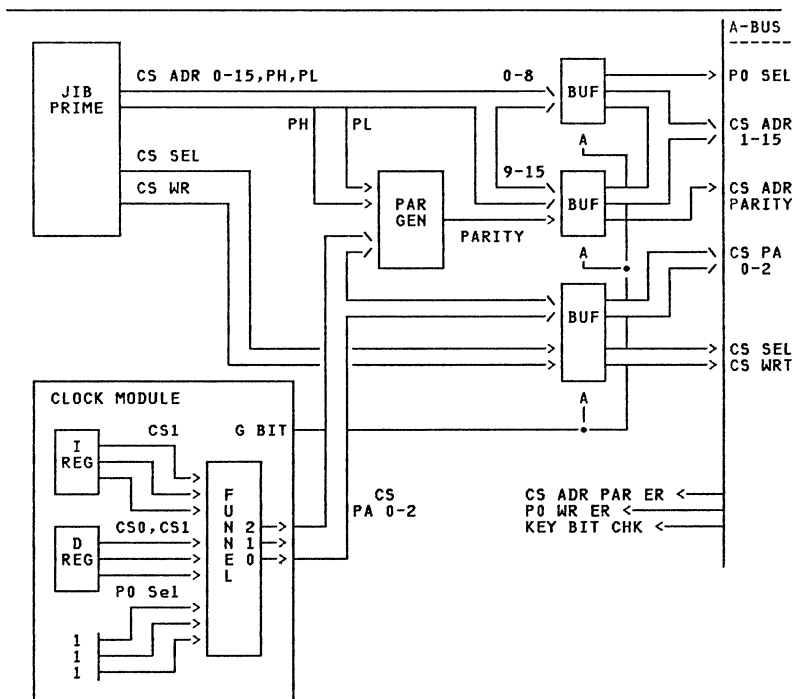


Figure 5. MEMORY ADDRESSING SCHEME: I reg -> Instruction register, D reg -> Data register. When Partition 0 is selected then the CS PA 0-2 bits are set to 1's. Also, G bit (any Cycle Steal Ack) when active, tri-states Hazel-A from memory.

Note: There are 8 sets of I and D registers. One for each Interrupt level.

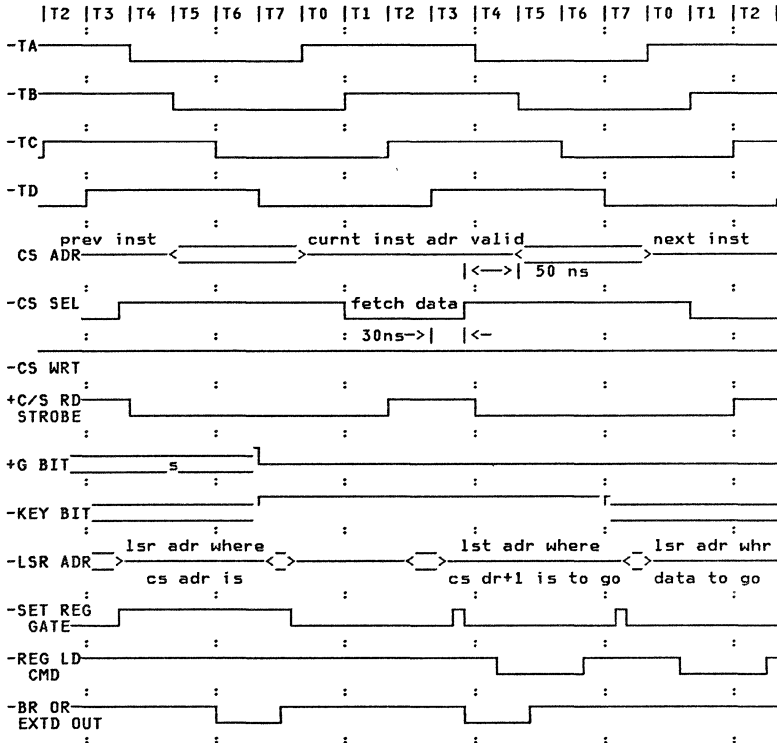
These registers can be accessed from the current interrupt level through XR X'18' and X'1A'. The D register may be modified by instructions located anywhere in memory. The I register should only be modified if the address of the next sequential instruction is between X'0000' and X'7FFF'. If the address of an instruction that modifies the I register is between X'8000' and X'FFFF', the I register will remain unchanged.

Note: With the exception of interrupt level 0, no level can access the I or D register of any other level. Interrupt level 0 uses XR X'1B' to determine which set of I and D registers appear in the XR space.

IBM Internal Use Only

Timing

The timings to fetch and store data from control store will be shown in the following diagrams. For further information concerning these timings see the JIB' FUNCTIONAL SPECIFICATION MANUAL.



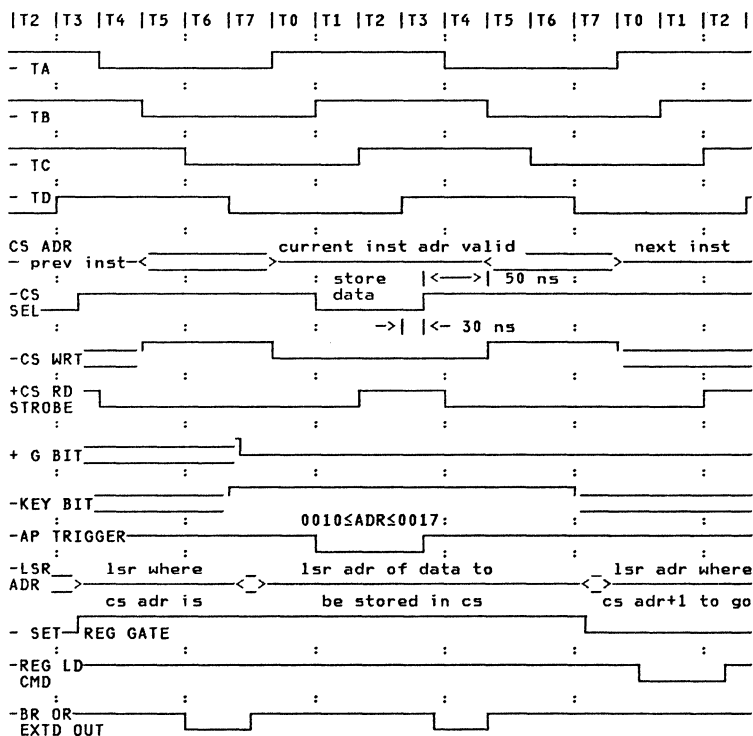
Note: The G-bit signal goes active during non-cycle steal operations to guarantee the A-bus is 3-stated when cs sel is inactive.

Note: Key Bit can change state at the beginning of T7 only. If Key Bit is gnd then A cycle steal. If Key Bit is +5v then a Hazel-A cycle.

Note: When accessing memory, T1 time will be stretched to 208.33 nsec.

Figure 6. Control Store Fetch Timing: All T times are 41.667 nsec except T1 during memory access. Also, where applicable, the timings shown are from the A-Bus.

IBM Internal Use Only



Note: The G-bit signal goes active during non-cycle steal operations to guarantee the A-bus is 3-stated when cs sel is inactive.: enp2.

Note: Key Bit can change state at the beginning of T7 only. If Key Bit is gnd then A cycle steal. if Key Bit is +5v then a Hazel-A cycle.

Note: When accessing memory, T1 time will be stretched to 208.33 nsec.

Figure 7. Control Store Write Timing: All T times are 41.667 nsec except T1 during memory access. Also, where applicable, the timings shown are from the A-Bus.

Because memory access is longer than the allotted time within the JIB' cycle, clock control lengthens T1 time of the cycle to 208.33 nsec allowing memory to be read or written.

Errors

There are 6 possible error conditions when accessing Hazel-A memory. They are categorized in two groups; errors occurring while writing Hazel-A memory and errors occurring while reading Hazel-A memory.

IBM Internal Use Only

• Hazel-A Memory Errors During STORE OPERATIONS

1. Control Store Data Parity Error
2. Control Store Address Parity Error
3. Partition 0 Write Error

• Hazel-A Memory Errors During FETCH OPERATIONS

1. Control Store Address Parity Error
2. Control Store Uncorrectable Error
3. Control Store ECC Error
4. Key Bit Check (not really an error)

These Errors generate a level 0 interrupt and are available through the Level 0 Interrupt register, XR '08'.

CYCLE STEAL INTERFACE

Level Definition

Cycle Steal Level	Description
0	Memory Refresh
1	Attached Device
2	Attached Device
3	Attached Device
4	Attached Device
5	Attached Device
6	Attached Device

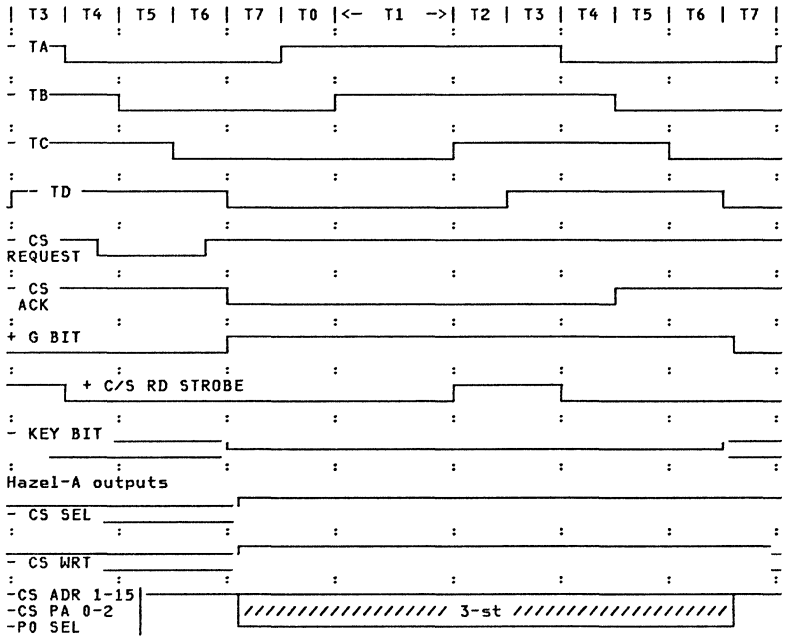
The Cycle Steal Interface allows attached devices, such as other processors, access to Hazel-A memory. Thus attached devices can easily read and write data from/to the Hazel-A engine. It should be noted that the attached device must generate it's own control signals to memory, except Cycle Steal Read Strobe and the Key Bit. These two signals will always be generated by Hazel-A.

During a Cycle Steal the Hazel-A JIB' clocks continue to run within the clock module. They are gated off in the clock module drivers by G Bit, thus JIB' is suspended for one complete cycle (T1 is always stretched to 208.33 nsec during a cycle steal). G Bit is the and of all the Cycle Steal Acknowledge lines. Thus if an attach device requests access to Hazel-A memory and Hazel-A acknowledges the Cycle Steal Request, then G Bit goes active and 3-states Hazel-A from the memory bus. The acknowledge to a Cycle Steal Request will go active at the beginning of T7. G-Bit will go active at the beginning of T4. This guarantees the A-bus to be valid only when CS Sel is active. The Cycle Steal Acknowledge will go inactive before the beginning of the next T5 time. G bit will go inactive before the beginning of the next T7 time. This time, from T7 to the next T5, is the available time for the Attached Device to access memory. During this time T1 will be lengthened to 208.33 nsec. This will allow the Attached Device time to fetch/store data, monitor the Cycle Steal Read Strobe, and free up the memory bus. At the next T5 time, Hazel-A will catch any new Cycle Steal Requests. Thus the Attached Device must free up Cycle Steal Request before T5 to guarantee that an acknowledge will not be granted.

The Key Bit signal determines if the cycle is a Hazel-A memory access or an Attached Device cycle steal. Key Bit can change only at the beginning of T7. If a cycle steal is in progress, Key Bit will be ground(logical 1). If no cycle steal is in progress, then Key Bit will be +5 volts(logical 0). See Figure 8 on page 17 for timings.

Note: Even though all Attached Devices can request a memory access every cycle (worst case), Hazel-A will only Acknowledge (give up) at most 50 % of it's cycles. This is controlled by the clock module circuitry on Hazel-A. Thus guaranteeing that Hazel-A can not be held off by the cycle steal interface.

IBM Internal Use Only



Note: These timings show Hazel-A granting an acknowledge immediately after receiving the C/S Request. The worst timings will occur when Hazel-A has an active C/S Req every cycle. Thus, an acknowledge will be granted every other cycle, reducing the Hazel-A process capability by 50 %.

Note: C/S Req can occur any time, as long as it is active during T5 and goes inactive (unless another C/S is required) before the next T5

Figure 8. CYCLE STEAL INTERFACE TIMINGS: When G BIT is active, the CS ADR 0-15, CS PA 0-2, P0 SEL, will be 3-stated from the A-Bus. CS Sel, CS Wrt, P0 WR EN, EPE, and EES will be held high. Also, Cycle Steal Request is latched every T5 time.

IBM Internal Use Only

AP TRIGGER

If the Hazel-A needs to send information to an attached device, but the attached device is not requesting a cycle, then Hazel-A writes to a location in memory which will cause AP TRIGGER to go active. These memory locations (X'0010' to X'0017) correspond to the slot address as shown below.

MEMORY LOCATION	SLOT ADDR	SLOT #
0010	'000'	8
0011	'001'	1
0012	'010'	2
0013	'011'	3
0014	'100'	4
0015	'101'	5
0016	'110'	6
0017	'111'	7

Note: logical 0 = + 5v and logical 1 = gnd AP trigger should be used by the Attached Device as an interrupt so as not to keep Hazel-A waiting. (ie. Hazel-A could be waiting for the Cycle Steal before continuing.)

JIB PRIME PROCESSOR

For Information of JIB', see the JIB PRIME FUNCTIONAL SPEC

The Module P/N is 6035431 (U2)

ARITHMETIC PROCESSOR UNIT

Hazel-A uses the INTEL Arithmetic Processor Unit (APU), 8231A, which is directly compatible with the AMD 9511. For complete description and details of the APU see the Intel "MICROSYSTEM COMPONENTS HANDBOOK", Micro-processors and peripherals Volume II, (page 5-219 to 5-228).

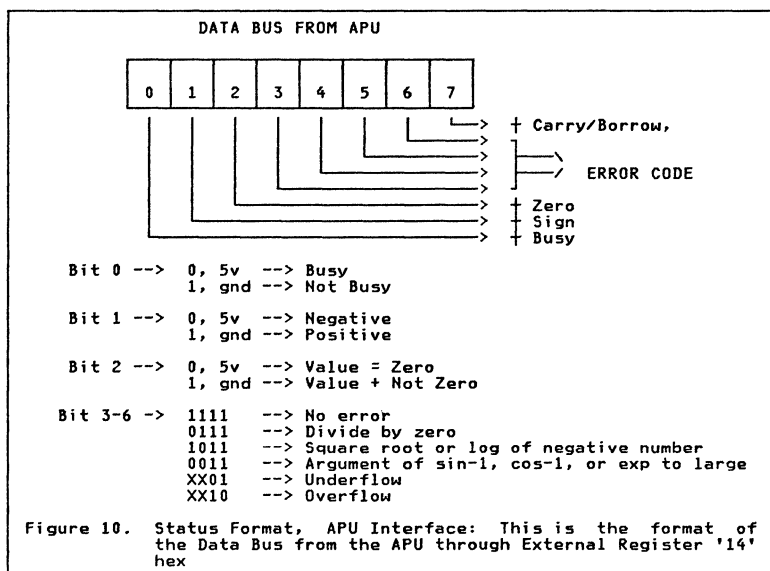
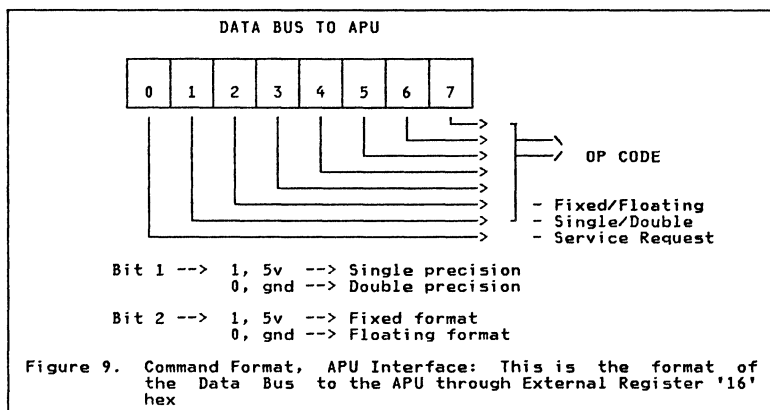
The APU provides the following functions.

1. Fixed point single and double precision (16/32)
2. Floating point single precision (32)
3. Binary data formats
4. Add, subtract, multiply, and divide
5. Trigonometric and inverse trigonometric functions
6. Square roots, logarithms and exponentials
7. Floating point to fixed point conversions and vice versa
8. Stack oriented operand storage
9. 8 Bit data bus interface
10. Standard 24 pin DIP package
11. + 12 volt and + 5 volt power supplies

Data Formats -- Command/Data

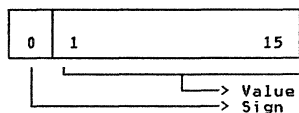
The APU data busses not only communicate data but also commands. The following diagrams describe the formats of the data, command, and status structures.

IBM Internal Use Only



IBM Internal Use Only

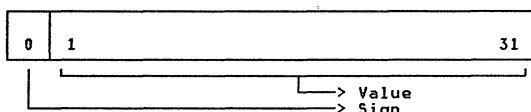
SINGLE PRECISION FIXED POINT FORMAT



Bit 0 --> 0, 5v --> negative number
1, gnd --> positive number

Bit 1-15 --> Range - 32,768 to + 32767

DOUBLE PRECISION FIXED POINT FORMAT



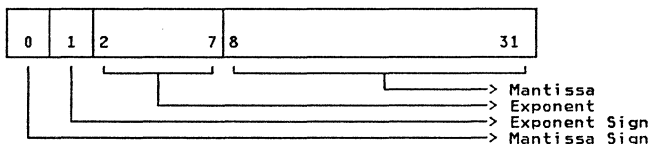
Bit 0 --> 0, 5v --> negative number
1, gnd --> positive number

Bit 1-31 --> Range - 2,147,483,648 to + 2,147,483,647

NOTE, THE VALUE IS ALWAYS EXPRESSED IN BINARY 2's COMPLEMENT

Figure 11. Data Format, Fixed Precision: This is the format of the Data Bus to/from the APU through External Register '14' hex and '16' hex

DOUBLE PRECISION FLOATING POINT FORMAT



Bit 0,1 --> 0, 5v --> negative number
1, gnd --> positive number

Bit 2-7 --> Range +/- 2.7x10**(-20) to +/- 9.2x10**(18)

NOTE, THE MANTISSA IS A FRACTION REPRESENTED IN BINARY

NOTE, THE EXPONENT IS REPRESENTED IN BINARY 2's COMPLEMENT

Figure 12. Data Format, Floating Point Precision: This is the format of the Data Bus to/from the APU through External Register '14' hex and '16' hex

IBM Internal Use Only

Interface to JIB PRIME

The interface to the external register bus of JIB' is accomplished through external register hex 14 to 17 (decimal 20 -23).

XR 14, hex -- 8 bit data bus from APU, bit 0 is MSB

XR 15, hex -- 3 bits used as control from the APU

BIT 0 --> (+) SVREQ ; Service Request, Signals command execution is complete and that post execution service was requested in the previous command byte
BIT 1 --> (-) READY ; Indicates that the APU is available to communicate across the bus
BIT 2 --> (-) END ; Indicates that command execution has been completed

XR 16, hex -- 8 bit bus to the APU, bit 0 is MSB

XR 17, hex -- 7 bits of control to the APU

BIT 0 --> (-) CS ; Chip select, Enables communications with the APU
BIT 1 --> (-) C/D (Ao); With the RD and WRT signals this control signal determines type of communication with the APU, control or data.
BIT 2 --> (-) RD ; Selects a read operation from the APU
BIT 3 --> (-) WRT ; Selects a Write operation to the APU
BIT 4 --> (+) RST ; Initializes the APU, does not affect the stack or command registers, and the APU is placed in an idle state
BIT 5 --> (-) SVACK ; Service Acknowledge, Clears the SVREQ signal
BIT 6 --> (-) EACK ; End Acknowledge, Clears the END of execution signal

APU Round-off Errors

When the APU is used to perform floating point functions, it is necessary that the programmer understand the round-off characteristics of the APU, and arrange his algorithms so that this has minimum impact.

Just because the APU has about 6 decimal digits of precision, it cannot be assumed that the result of a calculation also has 6 decimal digits of precision. The result could have very low precision under some circumstances.

For example:

999.998 - 999.988 gives the result 0.010. Due to the fact that the APU may round off the low order digit, the actual APU result could be anywhere from 0.0095 to 0.0105, a precision of 5% despite the fact that the precision of the numbers input to the calculation is 0/0001%.

CLOCK CONTROL

The clock module on Hazel-A handles all the timings to memory, Cycle steal interface, I, L, and D register banks, JIB' clocks and some miscellaneous control signals. The primary function of the clock module is to generate correct timing control, and addressing between JIB' and memory during fetch, store, cycle steal, and non-memory access.

listed below are some of the control signals and how they are generated by the clock module.

IBM Internal Use Only

Tomcat Clock ----> active during T1 always and when reading memory,
it is active during T5
Gate LSR to CS --> active when G Bit is off and when writing to memory
C/S Rd Strobe ----> active during T2 and T3 all the time
G Bit -----> active from T7 to T7 if a Cycle Steal
2 Mhz B & C clk -> generated via a 3 bit counter from the 24 Mhz
Oscillator input

Since the Clock Module has bits in all parts of Hazel-A, those parts of the clock module will be described in those sections. ie. the Cycle Steal control is described in the Cycle Steal Interface.

"Addressing" on page 12 will explain how the Control Store Page Address bits 0-2 (CS PA 0-2) are generated, and "Local Storage Registers TOMCAT" on page 30 will explain how the Extended LSR Address bits (LSR Adr bits 0-2) are generated.

The I, L, D registers (1 of each for 8 Int levels) are loaded from the DBD Odd bus bits 5, 6, and 7. The clocks for these registers are generated from control logic with the XR Adr, Mod Sel, Reg Ld Cmd, CS Sel, Br or Extd OP, CS Adr 0, G Bit, and the 8 interrupt levels which are decoded from the Interrupt level now active. The active interrupt level is generated either from the Extended Program SWap (EPSW - 3 bits) register or the Next Interrupt Level bits. If Level 0 is detected on the Next Interrupt Level bits, the EPSW register is used to select the level of I, D, and L registers otherwise the Next Interrupt Level bits are used.

Reading the I, D, or L registers is very much similar. The active Interrupt Level determines which set of I, L, and D registers to address. Thus 8 I regs, 8 L regs, and 8 D regs are funneled through separate multiplexers (lvl mux) selecting one of each to the Data bus funnel. The data bus funnel selects the proper register to the XR Data Bus (bits 5-7,P) and generates good parity. Figure 13 on page 23 shows this.

IBM Internal Use Only

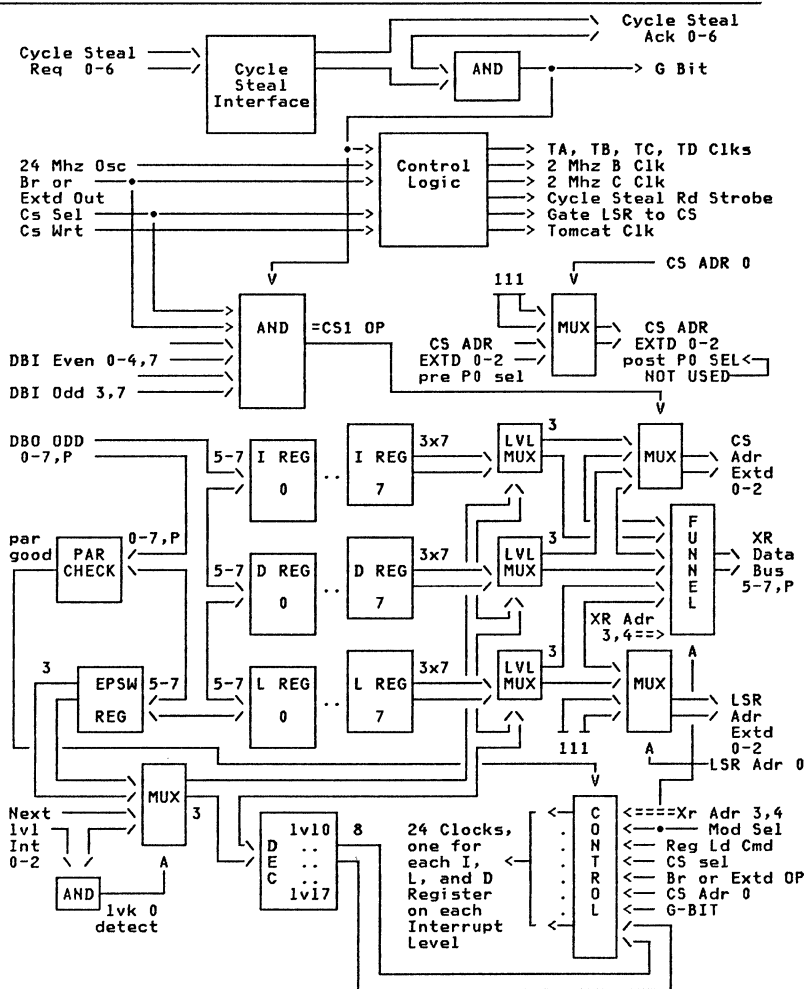


Figure 13. CLOCK MODULE 2nd LEVEL DIAGRAM: IBM P/N 8692020 (U20)

IBM Internal Use Only

HAZEL-A INTERRUPT THEORY

Level Definition

Int Level	Description
0	Machine Check, Attached Device
1	Timers APU
2	Attached Device
3	Attached Device
4	Attached Device
5	Software defined
6	Software defined
7	Software defined

The interrupt structure of Hazel-A can be broken down into 3 categories:

1. JIB' Interrupt Structure
2. Interrupt Handler Support
3. and Level 0 Interrupt

All three of these will be described in this section.

JIB' supports eight levels of interrupts. External hardware (the Interrupt Handler Module) catches interrupts and provides the new level which is prioritized and encoded to 3 lines. The interrupt is latched at T1 time under the following conditions:

- The current T7 is not in the first cycle of a 2-cycle instruction
- The current instruction is not an XI instruction to an XR address between X'18' and X'1F'. These registers belong to the Interrupt handler module, and when the handler is being addressed interrupts are masked off.

Once the interrupt is accepted, JIB' will save the old PSW (IAR, LSPR, Condition Codes, and Error codes, and the I, L, & D Registers) in the LSR's designated for that interrupt level (not the new interrupt level). JIB' will then turn on Interrupt Response for one full cycle. Thus guaranteeing that at least one instruction will be executed at the new level. The LSR definitions for each interrupt level are shown in "LSR Addresses for PSW (Int Level 0-7)".

LSR Addresses for PSW (Int Level 0-7)

Interrupt Level	LSR ADR
0	X'C0'-X'C3'
1	X'C4'-X'C7'
2	X'D0'-X'D3'
3	X'D4'-X'D7'
4	X'E0'-X'E3'
5	X'E4'-X'E7'
6	X'F0'-X'F3'
7	X'F4'-X'F7'

Note: The PSW values stored in these LSR's are the values at which the current level was entered, NOT the values at the current instruction.

Note: If the value of the PSW for the current level is modified, it will be overwritten when the level is exited.

Note: The PSW's for levels other than the current level are the values at which each level will start when entered. If they are modified, the corresponding levels will start with the modified values.

Note: There are also eight sets of I, D, and L registers, one for each interrupt level. With the exception of level 0, no level can access the

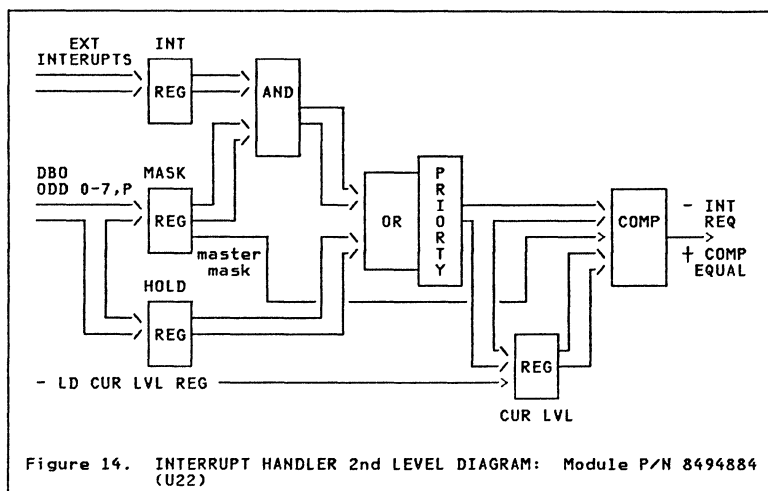
IBM Internal Use Only

I, D, or L registers of another level. Interrupt level 0 uses XR X'1B' to determine which set of I, D, and L registers appear in the XR space.

Note: The L and D registers can be modified by instructions located anywhere in memory.

Note: The I registers should only be modified if the address of the next instruction is between X'0000' and X'7FFF'.

Interrupt Handler



The Interrupt Handler module provides the masking, prioritizing, and interrupt request facilities for the JIB'. This circuitry determines the highest priority outstanding interrupt level that is not masked off. If this new level is different from the current level, the interrupt request line to JIB' is activated and the level is passed to JIB' on the interrupt level lines. JIB' uses its (CIL) Current Interrupt Level Register to generate the address to STORE the current PSW, and uses the Interrupt Level lines from the Interrupt Handler module to generate the address to FETCH the new PSW. At the completion of the PSW swap, JIB' sets the CIL register within itself to the value on the interrupt level lines and raises Interrupt Response. This line sets the external Current Interrupt Level register, in the clock module, to the new interrupt level and the compare network drops the Interrupt Request signal.

The first instruction of the microprogram at the new level should be to set the corresponding bit for this level in the interrupt level hold register. This bit will hold this interrupt level active for the duration of the microprogram, even though the original interrupt request is gone. The last instruction of the microprogram should turn off the corresponding bit for this level in the interrupt level hold register. This will cause a PSW swap back to the previously executing level. Since instructions which modify the interrupt level hold register cannot be interrupted, at least one instruction must be executed after the instruction which resets the level hold register, to allow PSW swapping.

IBM Internal Use Only

JIB' will always execute two cycles at the new interrupt level before another interrupt level switch can occur, as the interrupt cannot be raised until T3 time of the 2nd cycle following the PSW.

Level 0 Interrupt (machine check)

Level 0 interrupt is comprised of the following error conditions.

1. Control Store Address Parity Error
2. Control Store Data Parity Error
3. Control Store Uncorrectable Error (double bit)
4. JIB' Internal Error
5. Interrupt Level 0 from A-Bus
6. Key Bit Check

These signals are anded together and generate Interrupt Level 0 request. The memory errors CS DP ER, CS AP ER, CS UNC ER, and KEY BIT Check are masked off when a Cycle Steal Request has been acknowledged. Any one of these signals going active for a minimum of 25 nsec, will cause a level 0 interrupt. The Level 0 interrupt is reset by reading the Level 0 Interrupt Register (XR X'08').

IBM Internal Use Only

Timing

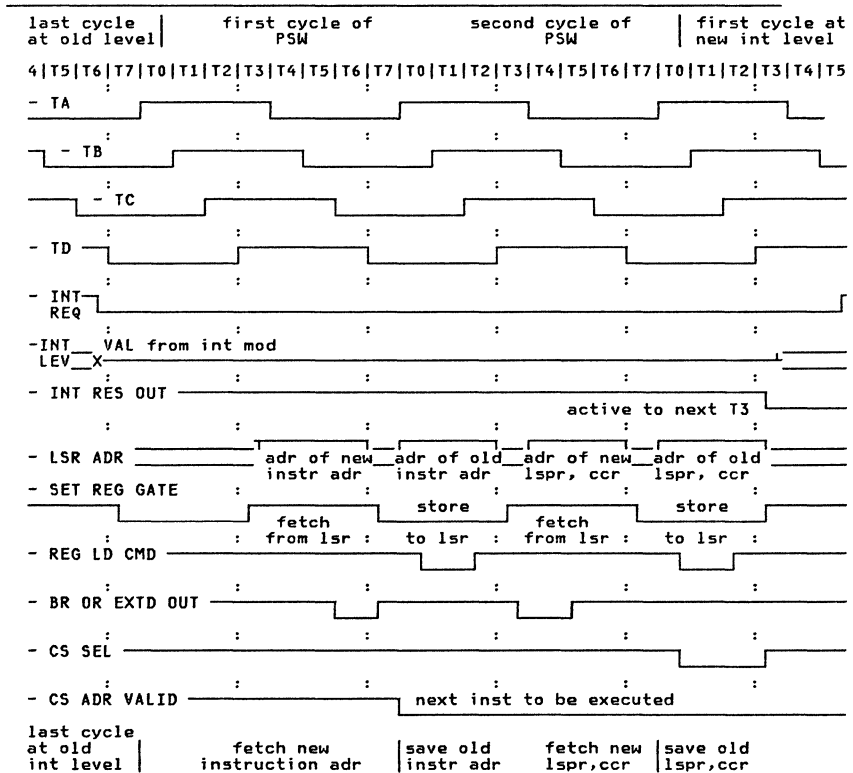


Figure 15. Interrupt Timing (PSW SWAP): All T times are 41.667 nsec

HAZEL-A TIMERS AND FREQUENCY DIVIDERS

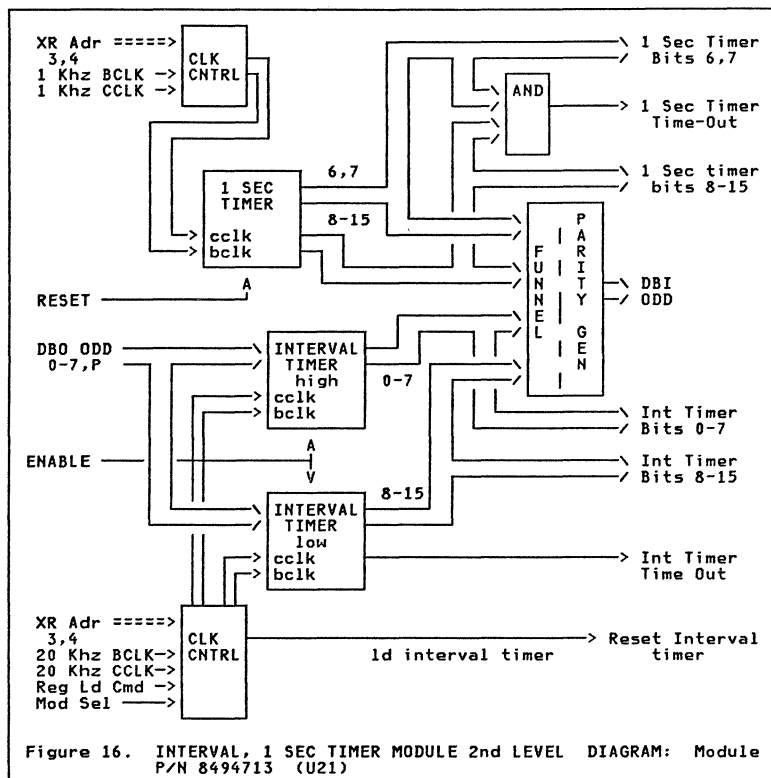
On Hazel-A there are two timers; Interval Timer and a 1 Second Timer. The One second timer is for time of day. The Interval timer can be used to implement delays, or waits in software. The Interval timer is loadable from the DBO ODD bus and the One Second timer runs without any interference.

These two timers cause level 1 interrupts along with the APU. These timers must be on level 1 or real time and accurate time-outs would not be available.

The 1 Second Timer runs off a 1 KHz clk. Thus every 1000 clk pulses (1 second) an interrupt is generated.

IBM Internal Use Only

The Interval timer runs off a 20 KHz clk. Thus, the least significant bit is worth 50 usec. The maximum time-out would be if the interval timer was loaded with hex 'FFFF'. This would give an elapsed time of 3.277 seconds ($65535 \times .00005$).

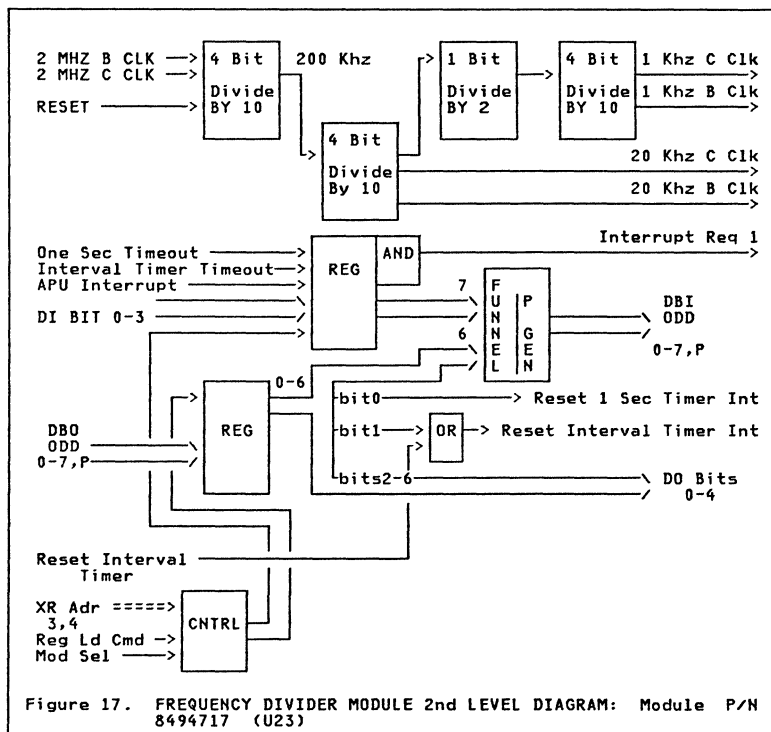


The Frequency Divider module generates the 2 clk frequencies for the Interval Timer module, 1 Khz and 20 Khz clks. It also contains a digital input register and a digital output register. 3 bits of the input reg are used to latch the level 1 separate interrupts (Interval and 1 sec time-outs and APU svreq). The remaining 4 bits are used as digital inputs. 2 bits of the output reg are used to reset the 1 sec and interval timers, and the remaining 5 bits are used as digital outputs.

Note: The input register does not use bit 7 and the output reg does not use bit 3. Thus only 7 of the possible 8 data bits are implemented.

The Interval timer is reset when either, the DO bit (reset interval timer) is turned on or the high or low byte of the timer is being loaded.

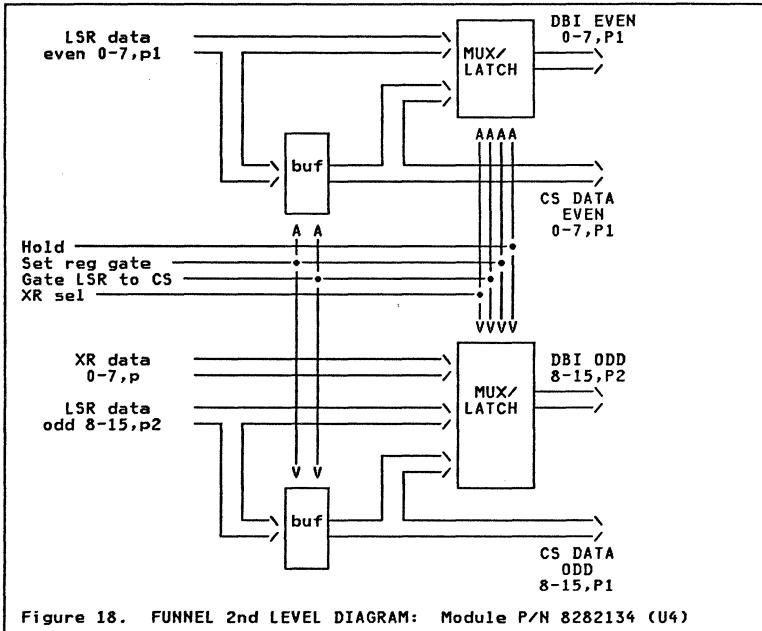
IBM Internal Use Only



FUNNEL

The funnel's primary functions are to funnel CS DATA or LSR DATA or XR DATA (for odd side only) to the DATA BUS IN to JIB', and also to gate LSR DATA to the CS DATA BUS.

IBM Internal Use Only



LOCAL STORAGE REGISTERS TOMCAT

The Local Storage Registers, LSR's are arranged in a 256x18 configuration. They will be used for local working registers, PSW's

IBM P/N for TOMCAT is 7378060
Document P/N for TOMCAT is 7379144

Timing

The LSR timing diagrams can be found in the JIB' Functional Spec.

Note: LSR's are written during T1 time and Read during T5 except during a CS0, CS1 Fetch. During these operations T5 time is used to store in LSR's the Next Instruction Address.

Addressing

IBM Internal Use Only

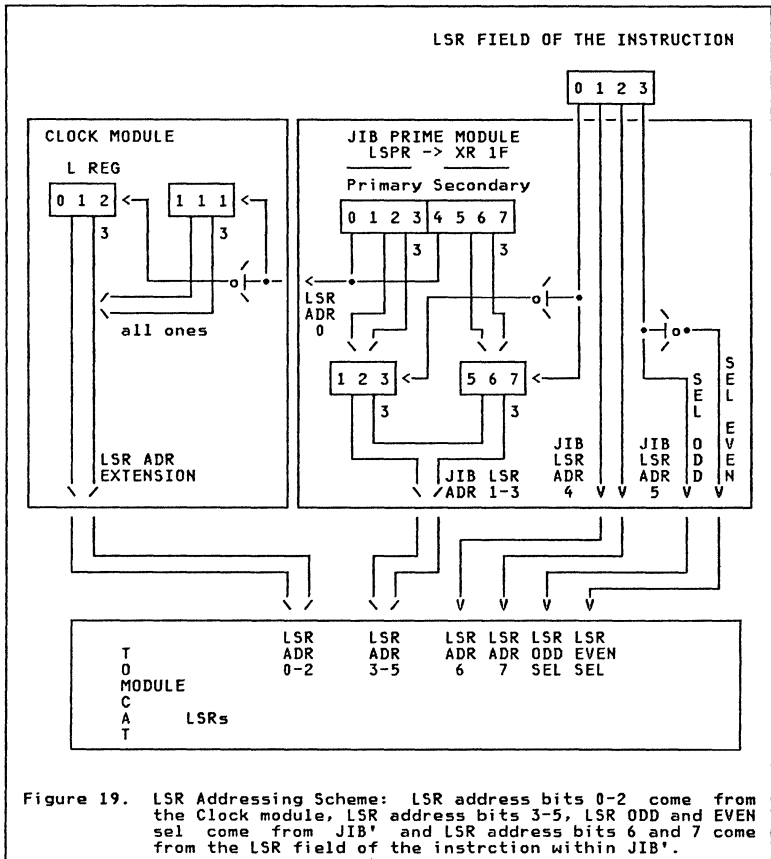


Figure 19. LSR Addressing Scheme: LSR address bits 0-2 come from the Clock module, LSR address bits 3-5, LSR ODD and EVEN sel come from JIB' and LSR address bits 6 and 7 come from the LSR field of the instruction within JIB'.

LSR addressing is extended over that of the JIB' by using the L register located in the clock module. Figure 19 illustrates how this is accomplished and how the complete address is generated.

The MSB of the LSR field, of the instruction, defines which half of the LSPR (Local Storage Page Register), located in the JIB', to use. The high order 4 bits of the LSPR is the Primary LSR Page and the low order 4 bits of the LSPR is the Secondary LSR Page. The reason for two 4 bit pages is to reduce the number of times writing the LSPR. Everytime you want a different page of LSR's you would have to write to the LSPR. Now you can set 2 pages and select between them through the LSR field of the instruction. The three LSB's of the Page register selected will be used as bits 3-5 of the LSR address. The MSB of the page register will be used to select the contents of the L register as bits 0-2 of the LSR address. If the MSB of the page register selected is 0, the contents of the L register is used. If the MSB of the page register is 1, bits 0-2 of the LSR address are set to 1's.

IBM Internal Use Only

Bits 1 and 2 of the LSR field of the instruction are used as bits 6 and 7 of the LSR address.

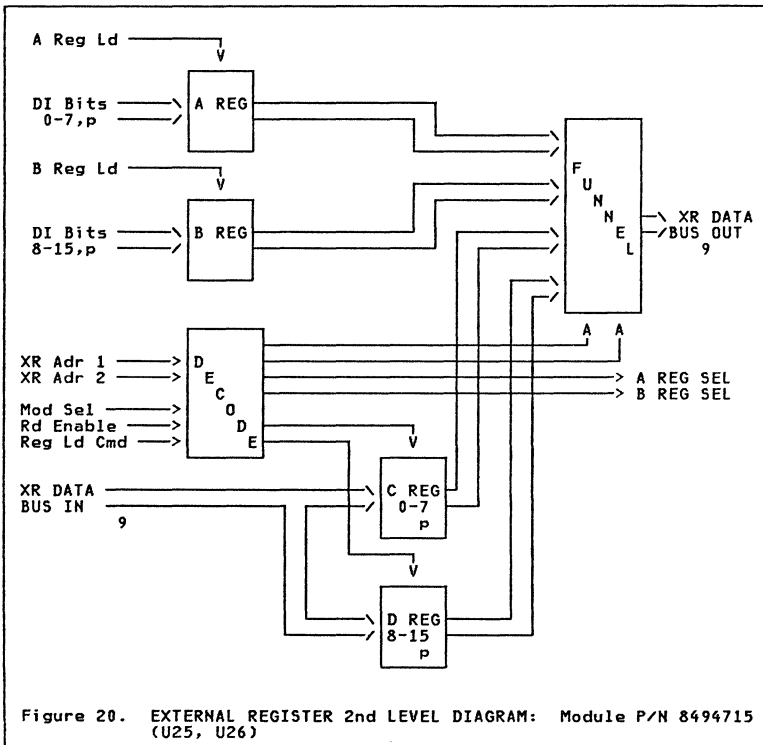
Bit 3 (the LSB) of the LSR field of the instruction is used as the even or odd select to the LSR module. If bit 3 is 0 then the LSR Even Select is active. If bit 3 is 1 then the LSR Odd Select is active. The LSPR bits select 1 of 8 boundaries.

Note: There are 8 sets of L registers. One for each Interrupt level.

The L register can be accessed from the current interrupt level through XR X'19'. The L register may be modified by instructions located anywhere in memory.

Note: With the exception of interrupt level 0, no level can access the L register of any other level. Interrupt level 0 uses XR X'1B' to determine which set of L registers appear in the XR space.

EXTERNAL REGISTERS



IBM Internal Use Only

Timing

Timing diagrams for external registers can be found in the JIB' Functional Spec.

Note: XR's are written during T1 time and Read during T5

Definitions and Addressing

The external registers are addressed via JIB'. The address consists of 5 bits plus parity, parity is not used. The first 3 bits (0,1,2) are decoded into eight group selects and the last 2 bits (3,4) select which of four registers to select in the group.

The External Register timing diagrams can be found in the JIB' Functional Spec.

Note: The External Registers are written during T1 time and are read during T5 time.

Note: The high order 3 bits of the address are decoded in the interrupt module.

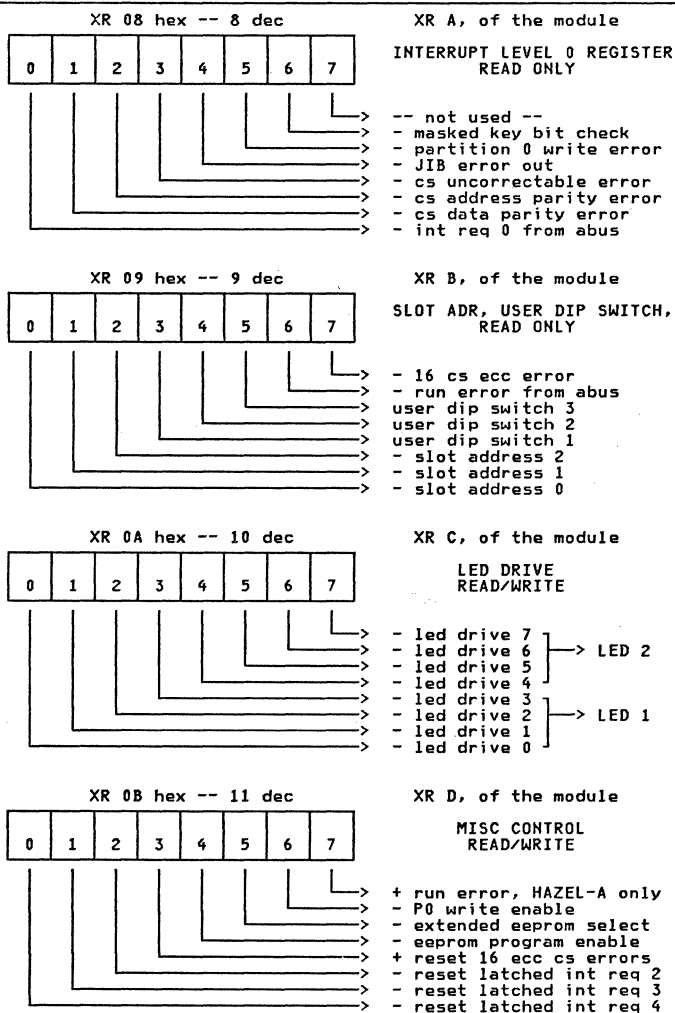
This section will break the registers into their respective groups and define the bits of each register. The Hazel-A utilizes 22 external registers from address '08' hex to address '1F' hex.

1. Group 0 --> not used
2. Group 1 --> not used
3. Group 2 --> see Figure 21 on page 34
4. Group 3 --> see Figure 22 on page 35
5. Group 4 --> see Figure 23 on page 36
6. Group 5 --> see Figure 24 on page 37
7. Group 6 --> see Figure 25 on page 38
8. Group 7 --> see Figure 26 on page 39

Note: The external register bit definitions are preceeded with a + or -. This defines the active state of that register bit.

- If the register bit is preceeded by a '+' then writing a logical 0 to that bit will put it in it's active state.
- If the register bit is preceeded by a '-' then writing a logical 1 to that bit will put it in it's active state

IBM Internal Use Only

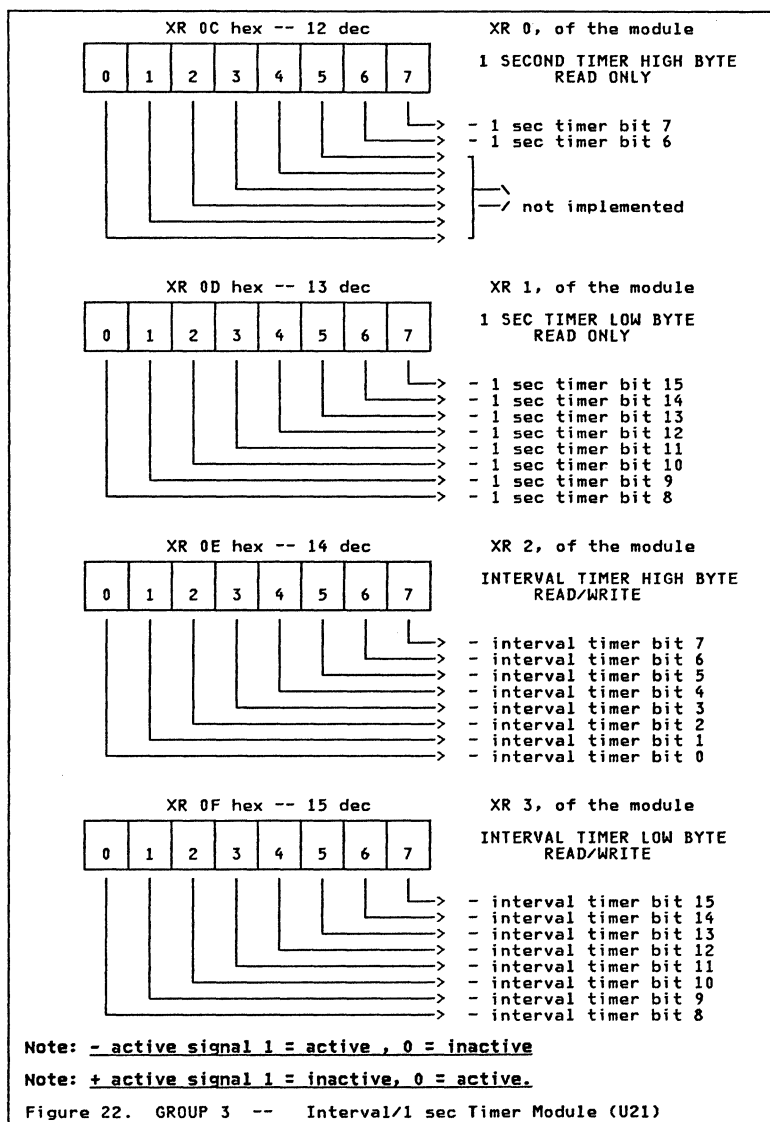


Note: - active signal 1 = active , 0 = inactive

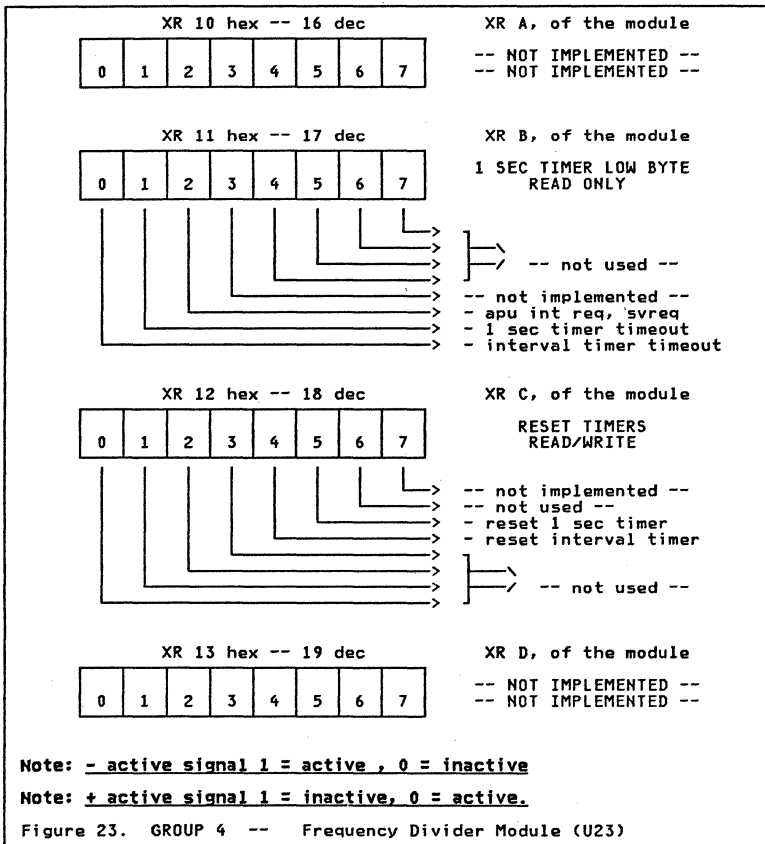
Note: + active signal 1 = inactive, 0 = active.

Figure 21. GROUP 2 -- External Register Module (U26): Miscellaneous controls, int level 0 register, errors

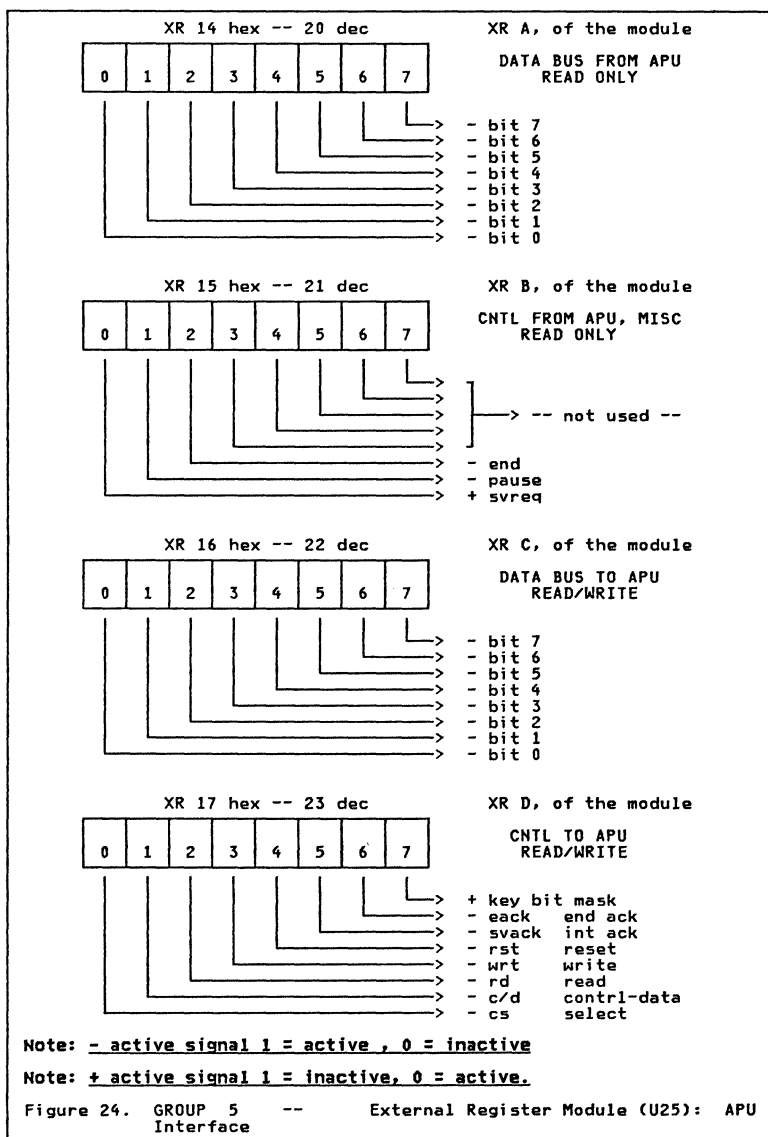
IBM Internal Use Only



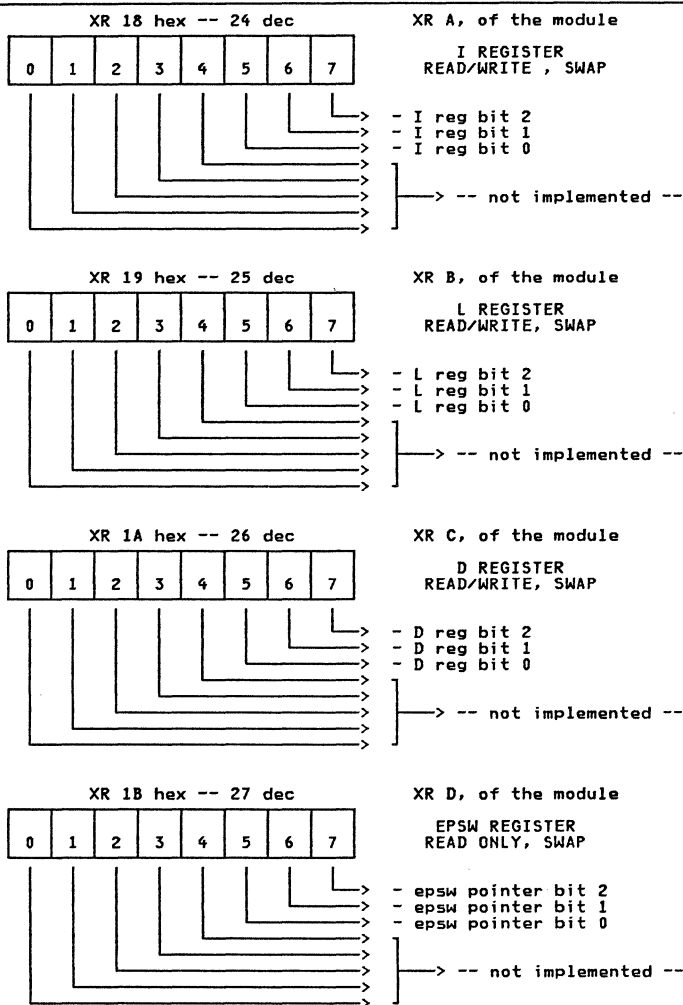
IBM Internal Use Only



IBM Internal Use Only



IBM Internal Use Only

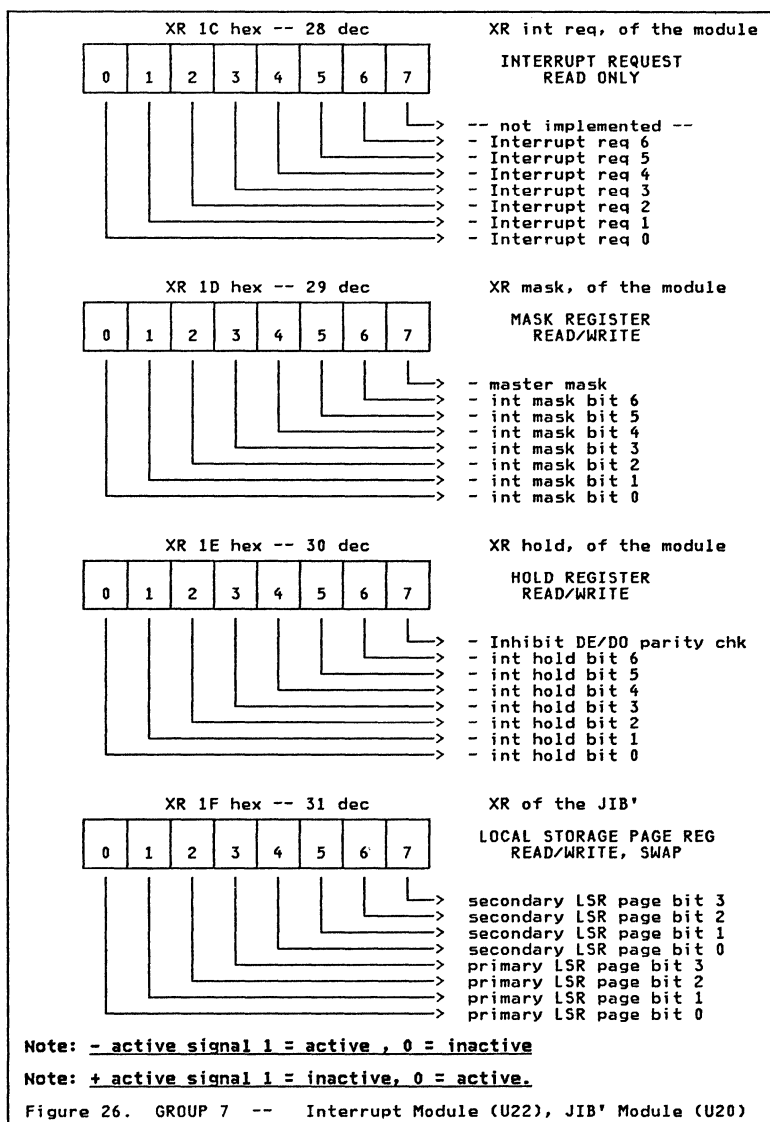


Note: - active signal 1 = active , 0 = inactive

Note: + active signal 1 = inactive, 0 = active.

Figure 25. GROUP 6 -- Clock Module (U20): I, L, D, EPSW Registers

IBM Internal Use Only



IBM Internal Use Only

HAZEL-IV TO HAZEL-A COMPARISON

1. Memory error checking
 - Control Store Data Parity Error
 - Control Store Address Parity Error
 - Control Store ECC Error (count of 16)
 - Key Bit Check
 - Partition 0 Write Error
2. Designed for Compatibility on the A-BUS
3. Enhanced Level 0 Interrupt
4. Interrupt Level 0, 2-4 are pulse triggered and Available on A-BUS
5. Easy Access to Test Points
6. 8 Diagnostic LED's
7. Run Error LED
8. 3 Hardware Programmable Switches
9. APU on Card
10. No DIS Interface
11. No Host Interface
12. No Keyboard Interface
13. No Maintenance Device

IBM Internal Use Only

APPENDIX A. TEST CONNECTORS

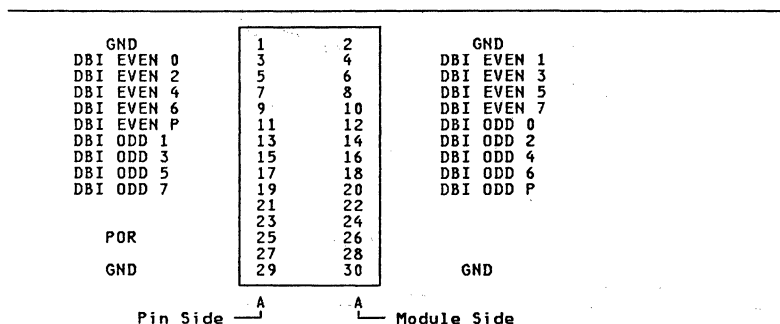


Figure 27. TEST CONNECTOR (JJ1)

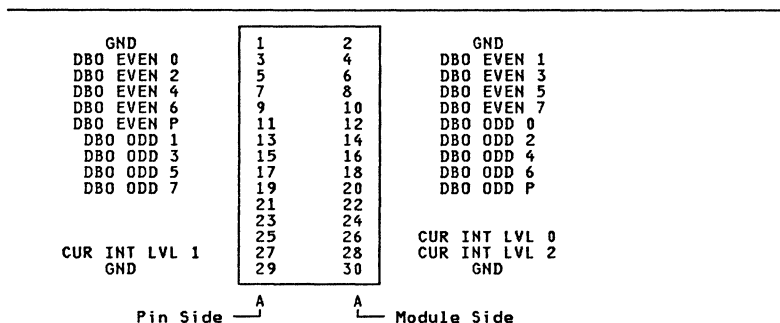


Figure 28. TEST CONNECTOR (JJ2)

IBM Internal Use Only

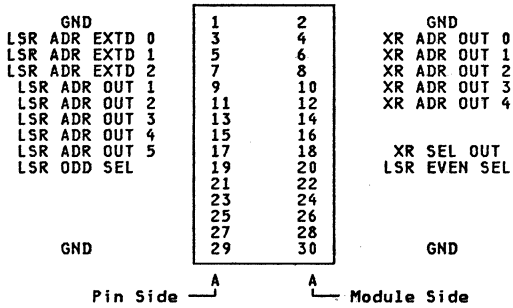


Figure 29. TEST CONNECTOR (JJ3)

IBM Internal Use Only

APPENDIX B. APU PROGRAMMING NOTES

Note: Not all APU OP CODES are supported. This is due to the support by EDX, The nucleus. See the software spec.

The following command sequence may be used to write data to the APU stack:

1. Set bits 0 (Select) and 1 (Data) in the APU Control Register (XR 17, hex)
2. Move the data byte into the APU Data out register (XR 16, hex)
3. Set bit 3 (Write) in the APU Control Register.
4. Wait for bit 1 (PAUSE) to drop in the APU Monitor Register (xr 15, hex)
5. Reset bit 3 in the APU Control Register
6. Repeat Steps 1 to 4 for each byte to be transferred.
7. Reset bits 0 and 1 in the APU Control Register.

The following command sequence may be used to read data from the APU stack:

1. Set bits 0 (Select) and 1 (Data) in the APU Control Register (XR 17, hex)
2. Set bit 2 (Read) in the APU Control Register.
3. Wait for bit 1 (PAUSE) to drop in the APU Monitor Register (XR 15, hex)
4. Read the data from the APU data in register (XR 14, hex)
5. Reset bit 2 in the APU Control Register
6. Repeat Steps 1 to 4 for each byte to be transferred.
7. Reset bits 0 and 1 in the APU Control Register.

The following command sequence may be used to read the APU status:

1. Set bit 0 (select) in the APU control register (XR 17, hex)
2. Set bit 2 (Read) in the APU Control Register.
3. Wait for bit 1 (PAUSE) to drop in the APU Monitor Register (XR 15, hex)
4. Read the status from the APU data in register (XR 14, hex)
5. Reset bit 2 in the APU Control Register
6. Reset bit 0 in the APU Control Register.

The following command sequence may be used to set the APU command register:

1. Set bit 0 (Select) in the APU control register (XR 17, hex)
2. Move the command into the APU data out register (XR 16, hex)
3. Set bit 3 (Write) in the APU Control Register.
4. Wait for bit 1 (PAUSE) to drop in the APU Monitor Register (XR 15, hex)
5. Reset bit 3 in the APU Control Register
6. Reset bit 0 in the APU Control Register.

The following command sequence may be used to determine if the APU has completed a command:

1. Set bit 0 (Select) in the APU control register (XR 17, hex)
2. Test bit 2 (End) in the APU monitor register (XR 15, hex) If the bit is off, the APU has not completed the preceeding command.

Note: It is advisable to have a timeout on this loop.

3. Reset bit 0 in the APU Control Register.

The following command sequence may be used to reset the APU:

1. Set bit 4 (Reset) in the APU control register (XR 17, hex)
2. Pause 2 microseconds
3. Reset bit 4 in the APU Control Register.

Heading ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Heading References</u>
card	HAZELA6	1	Card Overview
abusio	HAZELA6	6	A-BUS Inputs and Outputs
			12
memadr	HAZELA6	12	Addressing
			22
cstim	HAZELA6	14	Timing
cystl	HAZELA6	16	Cycle Steal Interface
			12
clk	HAZELA6	21	Clock Control
			12
intth	HAZELA6	24	Hazel-A Interrupt Theory
			6
psw	HAZELA6	24	LSR Addresses for PSW (Int Level 0-7)
			24
inthand	HAZELA6	25	Interrupt Handler
lsradr	HAZELA6	30	Local Storage Registers TOMCAT
			22
tc	HAZELA6	41	Appendix A. Test Connectors
			11

Figure ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Figure References</u>
card	HAZELA6	2	1: 3
hadata	HAZELA6	3	2: 3
habuss	HAZELA6	5	3: 6
hamem	HAZELA6	11	4: 12
haadr	HAZELA6	13	5: 12
csrd	HAZELA6	14	6: 16
cswr	HAZELA6	15	7: 16
cyst	HAZELA6	17	8: 16
apucmd	HAZELA6	19	9: 16
apust	HAZELA6	19	10: 16
apufx	HAZELA6	20	11: 16
apuf1	HAZELA6	20	12: 16
clk	HAZELA6	23	13: 22
int	HAZELA6	25	14: 22
inttime	HAZELA6	27	15: 22
timers	HAZELA6	28	16: 22
freq	HAZELA6	29	17: 22
fun	HAZELA6	30	18: 22
lsradr	HAZELA6	31	19: 31
xr	HAZELA6	32	20: 31
xrgr2	HAZELA6	34	21: 33
xrgr3	HAZELA6	35	22: 33
xrgr4	HAZELA6	36	23: 33
xrgr5	HAZELA6	37	24: 33
xrgr6	HAZELA6	38	25: 33
xrgr7	HAZELA6	39	26: 33
jj1	HAZELA6	41	27: 33
jj2	HAZELA6	41	28: 33
jj3	HAZELA6	42	29: 33

**ARIEL Project
Attached Processor
Source Communications Adapter (APSCA)
Functional Specifications**

Control Systems Engineering

**IBM Corporation
General Technology Division
D-035 B-300-47A
East Fishkill
tel 533-7654**

IBM Internal Use Only

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

ABSTRACT

The APSCA design is an adaptation of two previous designs.

1. The Attached Processor Card from the standard DSC IV
2. The Intelligent Source Comm. Adapter also from the standard DSC IV

These designs have been incorporated with a new firmware program to form the APSCA.

Additional logic circuits have been added in support of enhancements to the cycle steal (C/S) interface.

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

PREFACE

PURPOSE

This Functional Specification describes the Ariel Attached Processor/Source Communications Adapter (APSCA), a microprocessor driven adapter which will be used as the interface between the Hazel-A and the Remote DIS. Adapter commands will be fetched from Hazel Memory via the cards DMA interface and interpreted by the cards on board microprocessor. In addition, the card provides signals such as 'remote IPL', 'power sequence drive' and 'isolation drive' which are used by the tool.

RELATED DOCUMENTS

Related documents are:

- Remote DIS User Guide E45-1452 PN 6230664
- Distributed Sensor Subsystem IV Functional Spec E45-1289-0 PN 7834982
- Universal Block Interface Module (UBIM) Functional Spec E45-1454 PN 6230666
- Distributed Interface System, System Summary. Publication number Z45-0918
- Distributed Sensor Subsystem User Guide (Vol. 1). Publication number Z45-1123
- Distributed Sensor Subsystem User Guide (Vol. 2). Publication number Z45-1124
- Memory Components Handbook, Order Number 210830, Intel Corporation, 3065 Bowers Ave., Santa Clara, CA 95051
- EDXJ Programmers Manual

Refer all technical questions to Dept. 035, East Fishkill.

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

TABLE OF CONTENTS

General Description	1
Theory of Operation	1
Product Classification	1
Capacity	1
The following is a diagram of the architecture in which the	2
User Information (Hardware)	3
9 Pin "d" shell Connector	3
Coaxial Connector	5
ABUS Interface Connector	5
120 Pin Interface	5
20 Pin Interface	5
A-BUS Second Level Diagram	5
ABUS Pin Definitions	8
Test Interface	11
Test Interface Pin Assignments	11
User Interface (Software)	13
General	13
Block Interface Commands	14
TECHNOLOGIES	15
Printed Circuit Card	15
COMPONENTS	15
LOGIC	15
MEMORY	15
OTHER COMPONENTS	15
CONNECTORS	16
ORGANIZATION	17
Block Diagrams	17
APSCA Data Flow	18
REGISTER DEFINITION	20
LOCAL STORAGE REGISTERS	20
EXTERNAL REGISTERS	21
EXTERNAL REGISTER GENERAL DESCRIPTION	21
EXTERNAL REGISTER PARITY CONTROL	21
Communications	23
APSCA/Remote DIS Communications	23
General Operation	23
Transaction Format	23
Protocol	24
APSCA To RDIS Communications Interface	24
Line Characteristics	25
APSCA To LOCAL I/O Communications	25
APSCA to Hazel A Communications	25
APSCA to APSCA (peer to peer) Communications	26
CLOCK	31
JIB TIMING	31
Memory Selects	32
EXTERNAL REGISTERS	35
EXTERNAL REGISTER INTERFACE	35
OVERVIEW	35
ADDRESSING	35
DATA INTERFACE	35
CONTROL LINES	35
EXTERNAL REGISTER BIT ASSIGNMENTS	37
UBI Interface Registers	37
INTERRUPT STRUCTURE	49
GENERAL	49
Interrupt Level Definitions	49
APSCA Interrupts to Hazel A	50
THEORY OF OPERATION	50
INTERRUPT HARDWARE DESCRIPTION	51
OVERVIEW	51
INTERRUPT MODULE	52

IBM Internal Use Only

The Interrupt Request Register	52
The Interrupt Mask Register	52
The Interrupt Priority Tree	53
The Next Level Register	53
The Level Comparator	53
Interrupt Timing	53
ERROR CHECKING	55
GENERAL	55
JIB* DETECTED CHECKS	55
Errors during a Cycle Steel Operation	55
Machine Check Handler Requirements	56
DIAGNOSTICS	57
GENERAL	57
DIAGNOSTIC COVERAGE	57
DIAGNOSTIC LED CONTROL	57
MICROINSTRUCTION SET	59
MICROINSTRUCTION DESCRIPTION CONVENTIONS	59
EPROM PROGRAMMING	59
MICROINSTRUCTION SET DESCRIPTION	59
Firmware Specification	61
INTRODUCTION	61
HIGH LEVEL DESCRIPTION	61
ARCHITECTURE: LEVEL1/LEVEL3/LEVEL4	61
AP-DIS FUNCTIONS	62
RE-ISSUE HAZEL INTERRUPT	63
DIS I/O REQUEST	63
SERVICE LOCAL INTERRUPT	64
POLL NEXT BIC	64
MISCELLANEOUS ISSUES ASSOCIATED WITH INTERRUPT HANDLING	64
Reliability and Serviceability	67
Maintenance Support Plan	67
Environment	67
POWER SUPPLIES	67
Testing	68
Nodal Impedance Test	68
Functional Test	68
Required test Hardware	68
Required test Software	68

LIST OF ILLUSTRATIONS

Figure 1. Implementation of APSCA in Ariel Architecture	2
Figure 2. APSCA Card Outline	4
Figure 3. Remote IPL Interface	4
Figure 4. '9 Pin "d" shell user connector pin assignments'	5
Figure 5. 20 pin Interface Connector	6
Figure 6. ABUS Pin Assignments	7
Figure 7. 'Test Connector 1'	11
Figure 8. 'Test Connector 2'	11
Figure 9. 'Test Connector 3'	12
Figure 10. 'Test Connector 4'	12
Figure 11. 'RDIS Normal Block level Commands'	14
Figure 12. APSCA DATA FLOW	18
Figure 13. APSCA Address Control	19
Figure 14. 'Local Storage Register (LSR) Addressing'	20
Figure 15. 'External Register Assignment'	22
Figure 16. ARIEL Bus Local I/O Operations	26
Figure 17. ARIEL Bus Cycle Steal Interface Timing	27
Figure 18. Peer to Peer Communication Configuration	29
Figure 19. Peer to Peer Communications Timings	30
Figure 20. Clock Module Timings	33
Figure 21. 'External Register Interface Timing'	36
Figure 22. External Register Module	45
Figure 23. XR's of the Interrupt Module UI3	46
Figure 24. JIB' Local Storage Page Register	47
Figure 25. 'Interrupt Level Assignments'	49
Figure 26. 'Program Status Word (PSW) Format'	51
Figure 27. 'Interrupt Hardware'	54
Figure 28. 'Instruction Set'	60
Figure 29. HIGH LEVEL OVERVIEW OF AP-DIS FUNCTIONS.	62
Figure 30. AP-DIS INTERRUPT, DIS I/O AND COMMAND PROCESSOR.	63

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

GENERAL DESCRIPTION

THEORY OF OPERATION

The APSCA design is centered around a microprocessor which upon power on reset fetches and executes instructions contained in Erasable Programmable Read Only Memory (EPROM). The EPROM contains the firmware program which allows the microprocessor to perform APSCA functions such as:

- Communications to local and remote Distributed Interface System (DIS) blocks.
- Read/Write data via Direct Memory Access (DMA) to /from Hazel Memory.
- Handling of interrupts from DIS blocks and Hazel A engine.

Communications to the DIS is accomplished with the microprocessor interface to the Universal Block Interface (UBI) module. Serial data to/from this module are conditioned by an analog driver/receiver circuit which is directly connected to a coaxial cable.

The micro's of the APSCA and the Hazel-A have the capabilities of interrupting each other and communicate via Hazel Memory. A certain protocol will be followed when doing this and will be discussed in the firmware sections of this document.

The APSCA will function as an interrupt poller by addressing each RDIS block for interrupts. This makes all remote blocks look like they are directly attached. The APSCA will do this on command from the Hazel A when in the idle state.

User commands which are directed at the RDIS, originate in Hazel A via the EDX operating system which interprets high level EDXJ read/write commands.

The APSCA allows the HAZEL-A to communicate to another HAZEL-A located in the same ARIEL box via the common I/O bus which is part of the ABUS.

Good hardware performance is assured by bring up diagnostics which perform thorough checks of address/data and control lines and to all memory and external registers.

PRODUCT CLASSIFICATION

This product is intended for IBM's Internal Use Only.

CAPACITY

- Communications
 - Asynchronous
 - Single shielded coax cable (RG-62AU)
 - Multidrop up to 31 Remote DIS units
 - 1.25 MBAUD
 - Phase encoded signal
 - 6 bit CRC generation and checking on every transaction
 - Total cable length 5000 feet per drop
- DMA Interface
 - 16 Data Bits plus 2 parity bits
 - 19 Address Bits plus 1 parity bit
 - Hardware Handshaking
- Interrupt Capability

IBM Internal Use Only

- 4 Local I/O Interrupts
- 1 user interrupt
- 248 remote I/O interrupts
- 1 error interrupt
- 3 test and development interrupts

THE FOLLOWING IS A DIAGRAM OF THE ARCHITECTURE IN WHICH THE

APSCA will be used.

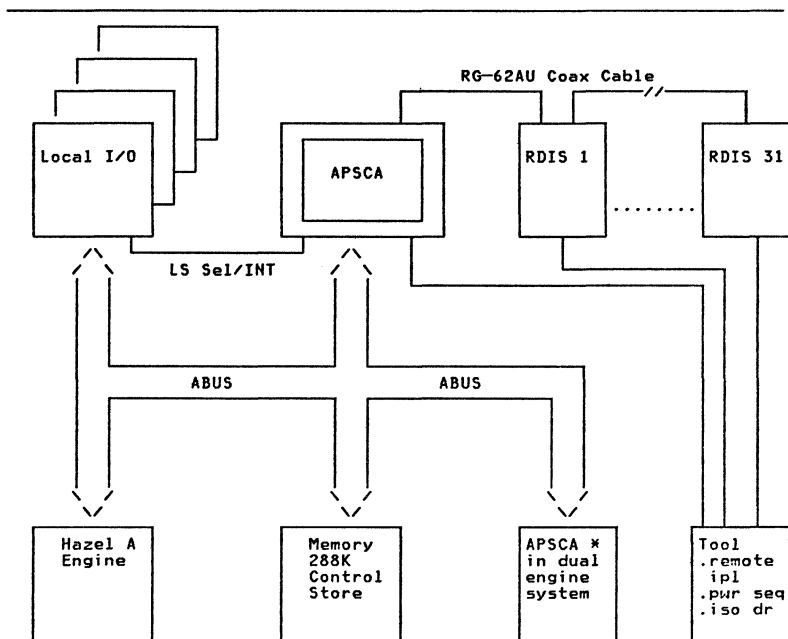


Figure 1. Implementation of APSCA in Ariel Architecture

IBM Internal Use Only

USER INFORMATION (HARDWARE)

This section provides information regarding the use of the card in the ARIEL system. The card is designed to be plugged into any of the 8 available sockets. If the Ariel box in which this card is located will be using local I/O cards then this card must be plugged into slot 8. It is recommended that the card be placed in this slot anyway because of it's additional 20 pin interface connector.

Certain complex applications may require more than 1 Ariel. In this case the system mother board will contain 2 ABUS's with 1 common I/O bus, this unique mother board will accept 2 APSCA's which must be located in slots 1 and 8. This configuration is known as a 'Dual Engine System' in which peer to peer communications is made possible via the common I/O bus. For further details see "APSCA to APSCA (peer to peer) Communications" on page 26.

9 PIN "D" SHELL CONNECTOR

This connector is used to interface to the Tool and provide the following functions.

User Interrupt. this should be a TTL compatible signal from the tool of any function required to interrupt the DSC. The DSC will be interrupted on both the up and down level of the signal. It is planned that the level of interrupt will be user programmable.

Note: This line is not supported in a dual engine system.

PSD Power Sequence Drive to the tool. When set to a logical '1' the signal goes minus. The signal is to be used to gate tool sensor/ actuator power on.

ISODR Isolation Drive. This is available to the user to drive isolation circuits or can be used as a general purpose line. This pin will go minus when set to a logical '1'.

USERBIT User Bit, can be used as a general purpose bit and will go minus when set to a logical '1'.

RIPL Remote IPL --- there are 2 pins provided so that Ariel can be IPL'd from up to 2 separate locations. These inputs are TTL and provide the necessary debounce circuitry so that switch contacts can be used. Isolation from the the ABUS is also provided. Bringing these inputs to a minus for a minimum of 100 milliseconds and then plus will cause the Ariel processors to reset thus causing an IPL. There is a logic ground return provided with each input. The pin assignments are shown in Figure 4 on page 5 and a block diagram is shown in Figure 3 on page 4.

IBM Internal Use Only

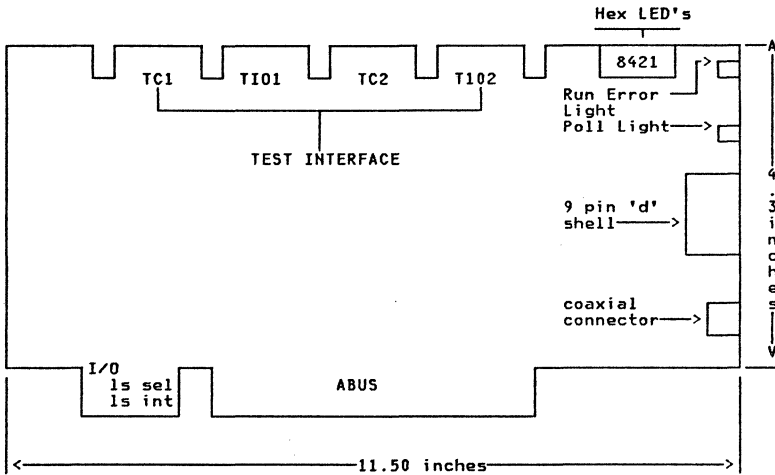
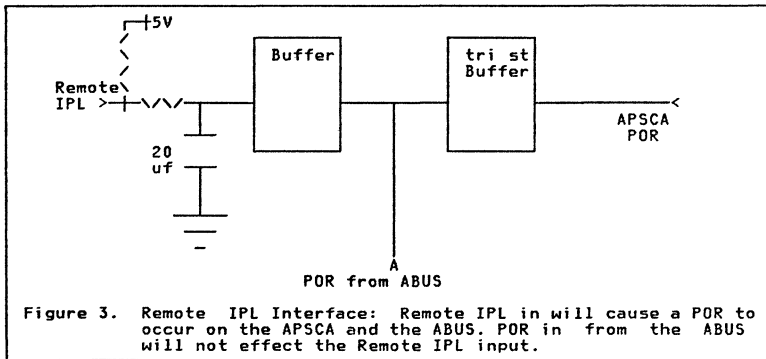
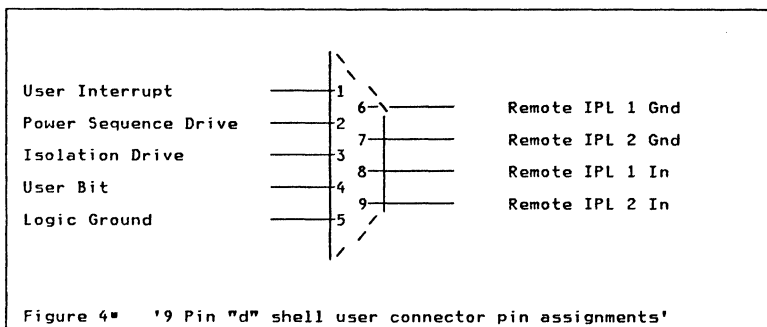


Figure 2. APSCA Card Outline



IBM Internal Use Only



COAXIAL CONNECTOR

This connector is intended to interface to coaxial cable type RG-62AU with an impedance of 93 ohms. See "APSCA/Remote DIS Communications" on page 23 for further details

Because of the 5/8" card spacing in the ARIEL box a sub-miniature (SMA) style connector is used on the printed circuit card. Since the outside world cabling features BNC style terminations an SMA to BNC adapter is used. This adapter will come standard with the APSCA. See "CONNECTORS" on page 16 for connector part numbers. Transmit and Receive data are connected to the center pin of the coax connector and the transmit and receive returns are connected to the shield.

ABUS INTERFACE CONNECTOR

120 Pin Interface

Located at the bottom of the card is a group of 120 contacts configured to mate with Amp p/n 5-530843-5. The APSCA utilizes the cycle steal interface, the power supply interface and the I/O interface which are all standard on the ABUS. For further details see "ABUS Pin Definitions" on page 8.

20 Pin Interface

Also located at the bottom of the card is a group of 20 contacts configured to mate with Amp p/n 5-530843-0.

This group is used in local I/O control and distributes the LS Select Lines and Interrupt Lines to the BIC. Figure 5 on page 6 is a diagram of the signals on the 20 pin connector.

A-BUS Second Level Diagram

IBM Internal Use Only

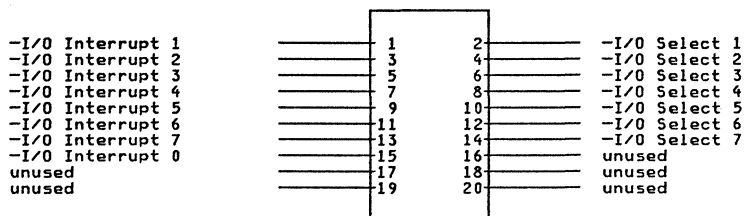
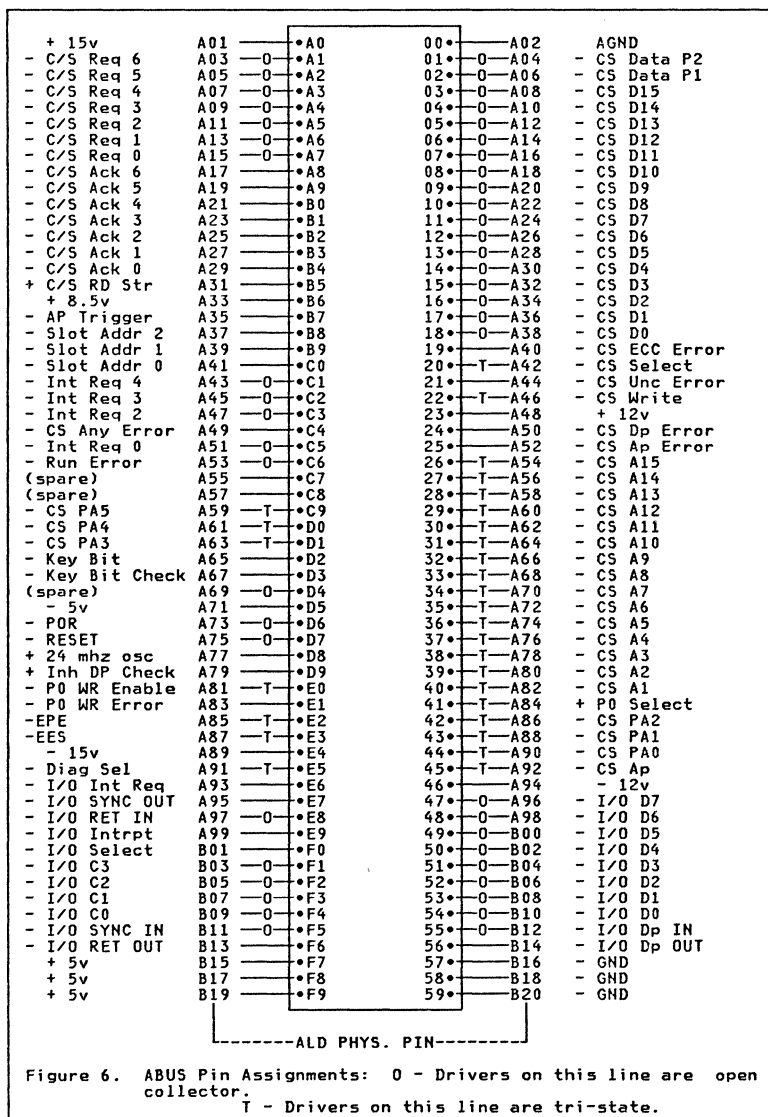


Figure 5. 20 pin Interface Connector

IBM Internal Use Only



IBM Internal Use Only

ABUS PIN DEFINITIONS

C/S -CS D0-D15	C/S Control Store Data Bus. Two bytes of data to/from Hazel A Memory.
C/S -CS Data P1-P2	C/S CS Data Parity. P1 and P2 are the odd parity bits for CS D0-D7 and CS D8-D15 respectively. These bits are provided by the APSCA via the CS Data Bus.
C/S -CS A1-A15	C/S Control Store Address (within a partition). A1-A15 define an address location within the 32K word partition boundary.
+P0 Select	C/S Control Store Address (partition 0 select line). Positive Active. When active, partition 0 of Hazel A Memory is selected regardless of the value of the CS PA0-PA2 lines.
C/S -CS PA0-PA2	C/S Control Store Address (partition select lines). These three lines select 1-of-8 partitions in Hazel A Memory when P0 Select is not active.
C/S -CS Ap	C/S Control Store Address Parity. Odd parity bit across the 19 lines which make up the control store address (CS A1-A15, CS PA0-PA2, and P0 Select). The APSCA creates this signal via a parity generator in which parity is generated across the 3 extended address bits and the P1 and P2 bits of the CS Address odd and even buses.
C/S -CS Select	C/S Control Store Select. Active during a Hazel A memory read/write operation. Additional VTL logic will generate this signal on the leading edge of Hazel A C/S Acknowledge. The signal is reset on the trailing edge of the Hazel A C/S Read Strobe.
C/S -CS Write	C/S Control Store Write. When low, data is written to Hazel A Memory. When high, data is read from Hazel A Memory. This signal is gated by Hazel A C/S ACK on the APSCA.
-CS ECC Error	Control Store Read Error. Active during control store read cycle if ECC logic on the memory card detects a single or double bit error. This line is readable through external register x'B' on the APSCA.
-CS Unc Error	Control Store Uncorrectable Error. Active during a control store read cycle if ECC logic on the memory card detects a double bit error. This line is readable through external register x'B' on the APSCA.
-CS Dp Error	Control Store Data Parity Error. Active during a control store write cycle if Hazel A Memory logic detects bad parity across CS D0-D15. When low, this line inhibits a write to Hazel A Memory. This line is readable through external register x'B' on the APSCA.
-CS Ap Error	Control Store Address Parity Error. Active during a control store read/write cycle if Hazel A Memory logic detects bad parity across the control store address (CS A1-A15, CS PA0-PA2, and P0 Select). When low, this line inhibits a write to control store. This line is readable through external register x'B' on the APSCA.
-CS Any Error	Hazel A Memory Any Error. Logical OR of all the error lines. Will cause a level 0 interrupt on the APSCA during a cycle steal operation only.
-P0 WR Enable	Partition 0 Write Enable. When low, partition 0 is not write protected. When high, address locations X'0000' to X'5FFF' in partition 0 are write pro-

IBM Internal Use Only

tested. This line is controlled via external register X'C'.

- P0 WR Error** Partition 0 Write Error. Active during a write operation to a protected section of partition 0. This line inhibits a write to control store. This line is readable through external register x'B' on the APSCA.
- Key Bit Check** Key Bit Check. Although this line is wired on the APSCA it is not supported in ARIEL and thus will not be supported by the APSCA.
- C/S Req 2-5** HAZEL-A Cycle Steal Request. 4 prioritized cycle steal request lines to the clock control logic on the HAZEL-A card. C/S Req 2 being the highest priority. C/S Req 2,3,4,5 are the only request lines supported by the APSCA and are controlled via external register x'C' bits 6 and 7 as follows.

Bits 6 7 -C/S Req

0 0	2
0 1	3
1 0	4
1 1	5

- C/S Ack 2-5** HAZEL-A Cycle Steal Acknowledge. This is an indication by the Hazel-A that it is alright to access Hazel-A memory. The APSCA then uses this line to gate address and data onto the cycle steal bus. C/S Ack 2,3,4,5 are the only lines supported by the APSCA and are controlled via the same bits that the Cycle Steal Request lines are.
- +C/S RD Str** Cycle Steal Read Strobe. Positive Active. This line is active at any time the data from Hazel A Memory is valid, regardless of whether cycle steal is active. The APSCA uses this line to reset CS Select and to latch the data in the data funnell module.
- Int Req 0,2-4** HAZEL-A Interrupt Request. 4 prioritized interrupt request lines to the interrupt handler logic on the HAZEL-A card. Int Req 0 being the highest priority. Only one of these lines will be used by the APSCA and it is programmable through XR 'C' bits 4,5. The select table is as follows.

Bits 4 5 -Int Req

0 0	0
0 1	2
1 0	3
1 1	4

- AP Trigger** Attached Processor Trigger. This line goes active when the HAZEL-A card writes into memory locations in the range X'00010' to X'00017' of partition 0. It is used in conjunction with the trigger word decode logic of the Interrupt Handler Module, the trigger word is determined by the slot address. A level 1 interrupt to Jib' will occur when the Trigger Word matches the Slot Address.
- Slot Addr 0-2** Slot Address. These 3 lines are prewired on the mother board and define the slot number, 0 through 7. They are wired into the Interrupt Handler module and are used in conjunction with the trigger word and the AP Trigger. They are also readable via external register X'1B'. These lines are negative logic convention and a logical '0' = +5v; and a logical '1' = GND.

IBM Internal Use Only

-RESET	HAZEL-A Reset. When this line is held low, HAZEL-A is reset. When the line goes high, HAZEL-A starts executing from location X'00000' of partition 0. The APSCA controls this line via XR '1B' bit 0. This line will not be supported for this application but has been implemented in lieu of possible future applications.
-POR	Power On Reset. When this line goes low the APSCA is reset to a known state. When the line goes high, the processor begins execution at location X'0000'. This line can also be activated via the Remote IPL lines provided on the 9 pin 'd' shell user connector.
-Run Error	Run Error. This line is brought low by the APSCA to indicate that it is not running properly or that it has not yet finished bring-up procedures. The APSCA controls this line via XR 'C' bit 0.
-I/O D0-D7	Bidirectional I/O Data Bus. 8-bit bidirectional data bus to/from any local I/O cards up to 4 cards may be plugged into the local bus.
-I/O Dp OUT	I/O Data Parity Out. Odd parity bit for I/O D0-D7 from APSCA logic to local I/O cards, valid during an I/O write operation. If the cards detect an error they will ignore the transaction.
-I/O Dp IN	I/O Data Parity In. Odd parity bit for I/O D0-D7 from macrofunction cards to APSCA logic, valid during an I/O read operation if the Inhibit Parity Gen Line of the UBI module is active.
-I/O C0-C3	I/O Command Tag Bus. 4-bit command tag bus to the macrofunction cards.
-I/O SYNC OUT	I/O SYNC Out. Control line generated by APSCA logic to initiate a transaction on the I/O bus. When low, data and command tag information is valid for processing by the currently selected macrofunction card. In a dual engine system this line is wired to I/O interrupt 0 of the other APSCA card. The line is activated when it is desired to transfer data from one DSC to another.
-I/O RET IN	I/O RETURN In. Control line generated by local I/O cards to signal the successful completion of a transaction on the I/O bus. When low, return data is valid for processing by the APSCA logic. This line is also used in a dual engine system by the other APSCA to indicate it has received the data transfer successfully.
-I/O RET OUT	I/O RETURN Out. Used in dual engine system and wired to the I/O Return In of the other APSCA, activated to indicate that data has been received successfully from the other APSCA.
-I/O Intrpt	I/O LSA Interrupt. This line is wired to the User Interrupt on the 9 pin 'd' shell user connector. In a single engine system the mother board connects this line to I/O Interrupt 0 of the 20 pin connector, in a dual engine system this line is not supported as Int 0 of the 20 pin connector is used for peer to peer communications. This line does not have a predefined active state, an interrupt occurs when the state of this line is different than the state of the corresponding bit in the BIC Reference Register.

IBM Internal Use Only

TEST INTERFACE

Located at the top of the card are 4 groups of 30 contacts configured to mate with Amp p/n 499186-4 or 530843-7 and have the designations TCI 1&2 and TIO 1&2. Test inputs are brought into the TCI's while input/outputs are brought to the TIO's.

Test Interface Pin Assignments

The next 4 figures contain the test pin assignments.

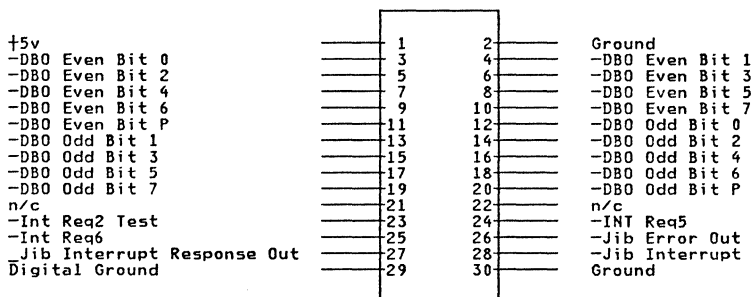


Figure 7. 'Test Connector 1': 'This connector is designated as TCI1 on the ALD p/n 6912762'

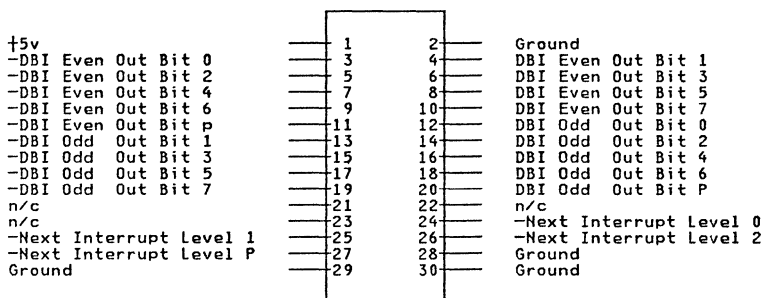


Figure 8. 'Test Connector 2': 'This connector is designated as TIO1 on the ALD p/n 6912762'

IBM Internal Use Only

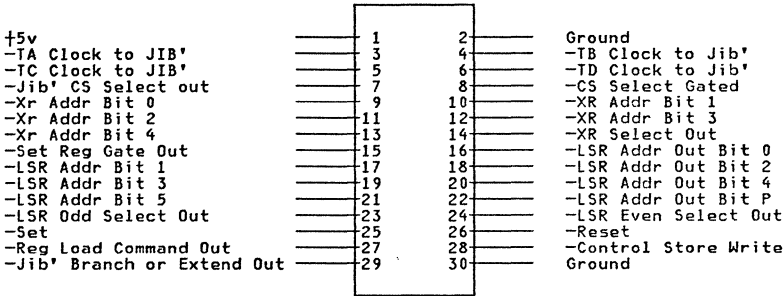


Figure 9. 'Test Connector 3': 'This connector is designated as TC12 on the ALD p/n 6912762'

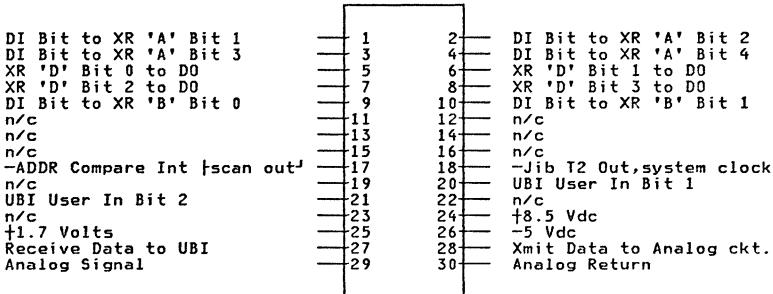


Figure 10. 'Test Connector 4': 'This connector is designated as TI02 on the ALD p/n 6912762'

IBM Internal Use Only

USER INTERFACE (SOFTWARE)

GENERAL

The EDXJ user programmer will perceive the APSCA as being transparent to all the Remote blocks of DIS and as a block interface (BIC) for any local I/O. The standard DISWRITE/DISREAD commands are still the mechanism for communicating to the sensor I/O. Interpretation of the commands at the nucleus level however, is handled differently and only an ARIEL Nucleus should be used in the system.

For further details on the EDXJ commands see the EDXJ Programmers Manual.

IBM Internal Use Only

BLOCK INTERFACE COMMANDS

As mentioned the APSCA represents 1 block of local I/O and up to 31 blocks of remote I/O. In either case the following block interface command sequences are supported.

CTC	OPERATION	BIT FUNCTION
0	Address only Poll for Interrupts	
1	not supported	
0 +3	Write Reference Register. Interrupt generated on unequal compare	8-15 = Interrupts
0 +5	Read Interrupt Register	8-15 = Interrupts
0 +7	Read Reference register. Interrupt generated on unequal compare	8-15 = value of reference register
0 +9	Read Control Register	8 = Interrupt Enabled 9 = Power Sequence to System power Creg 1 10 = Not used (0) 11 = Isolation Drive Creg 3 12 = Not used (0) 13 = User Bit Creg 5 14 = Interrupt Pending 15 = Watch Dog Enable Note: When writing a logic 1 in this bit, places a neg. level on user pin
0 +B	Write Control Register	Same as CTC 9
0 +D	Write Interrupt Register	8-15 = Interrupts
0 +F	Read Macrofunction ID	0-7 = Not used (0) 8-15 = ID = 01

Figure 11. 'RDIS Normal Block level Commands'

IBM Internal Use Only

TECHNOLOGIES

PRINTED CIRCUIT CARD

The APSCA components and connectors are packaged on a planar style card. The card will have 6 signal planes and a voltage and ground plane wired two lines per channel. The card requires supplies of +5, +8.5 and -5 volts.

The card will be documented and released with the following part numbers:

- Assembly pn.6912760
- Raw Card pn.6912761
- Schematic pn.6912762
- E.C. number A37493

COMPONENTS

LOGIC

The APSCA uses the JIB' microprocessor module, a 28mm module with 116 pins, of which 94 are signal I/O. In addition, the UBI (universal block interface) module is used. Both modules are granite technology.

All other logic on the card is implemented in GOLF (MS-438) technology, and in VTL.

MEMORY

The Local Storage Registers (LSR) for JIB' are provided by one SAMANTHA (MS-615) module (P/N 8041469). The LSR is addressed as either 64 halfwords or as 128 bytes.

JIB' control storage is implemented in two 8Kx8 Vendor EPROM's (INTEL 2764A-25) and three 2Kx8 static RAM's. Memory is accessed as 18 bits wide (16 data bits and 2 parity bits). Parity is not generated by the EPROM's, so external VTL parity generators are used to prevent JIB from generating a parity error when accessing them.

The EPROM's occupy JIB' memory space between locations X'0000' and X'1FFF'. The RAM memory is in locations X'2000' to X'27FF'.

OTHER COMPONENTS

The oscillator is a 20 MHz crystal controlled VTL compatible circuit, vendor supplied, packaged in a TO-8 can.

The +1.7 Volt supply for the JIB' and UBI module is provided by an on-card regulator using a ZIRCON technology regulator, P/N 5616162, and discrete components. The output is temperature compensated.

Six Light Emitting Diodes (LED's) are provided as indicators.

Decoupling capacitors, resistors, transistors, and R-packs, are IBM standard items.

IBM Internal Use Only

Additionalal discrete transisitors, transformers and other passive components are used for the communications drivers and receivers.

CONNECTORS

1. User Interface Connectors
2. Mother Board Interface
3. Test and Diagnostics

The user interface connectors consist of one 9 pin 'd' style (IBM 6150680) connector and one 'sma' style for connection to the coax cable. A 'sma' to 'bnc' adapter is used to maintain compatibility with existing factory cable runs. The SMA connector is an Amphenol p/n 901-143 and the SMA adapter is an Amphenol p/n 901-166.

Interface to the mother board is accomplished through a 120 pin and a 20 pin contact arrangement laid out to mate with AMP 5-530843-5 and 5-530843-0 standard edge style connectors.

Connections for test and diagnostics is accomplished by four groups of a 30 pin arrangement of contacts designed to mate with AMP 499186-4 or 530843-7. The connections are made on the top of the card.

IBM Internal Use Only

ORGANIZATION

BLOCK DIAGRAMS

The APSCA data flow is shown in Figure 12 on page 18, the APSCA Address and Timing Control is shown in Figure 13 on page 19. For further information on the JIB' module refer to the JIB' Functional Specifications dated August 7, 1978.

IBM Internal Use Only

APSCA Data Flow

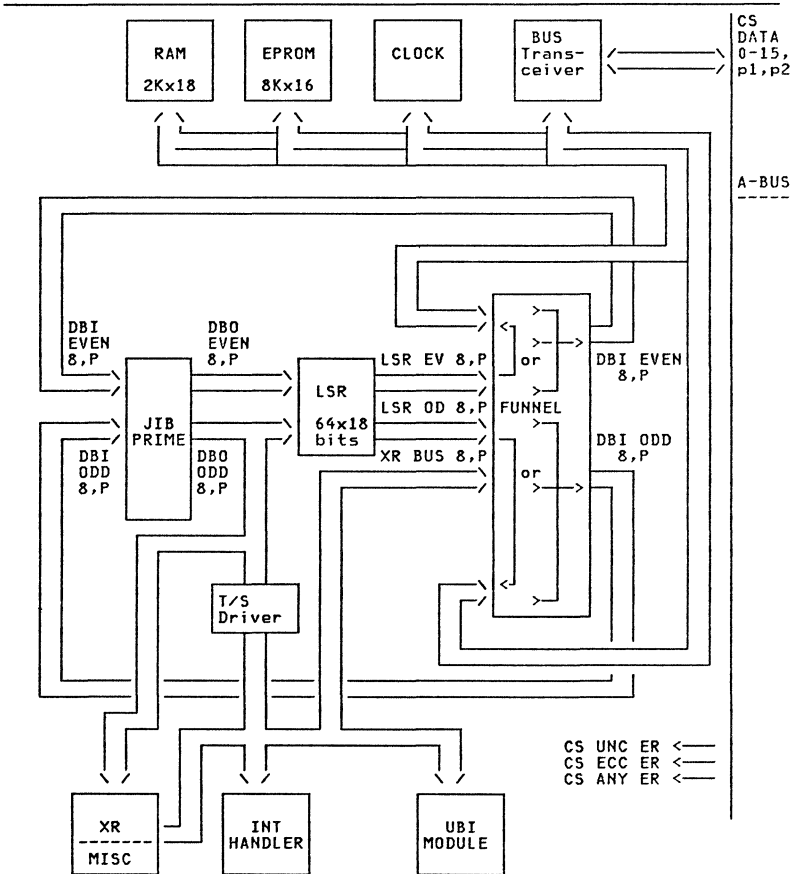


Figure 12. APSCA DATA FLOW

The diagram illustrates the JIB* system architecture. At the top, three modules (XREG, UBI, and Intrpt) provide address signals (A) to the XREG ADDR 3,4 and XR ADDR 0,1,2 inputs of the JIB* module. The JIB* module also provides signals to the P-GEN module (P2, P1) and the C/S Interface module (3, 16). The C/S Interface module is connected to the ABUS and AP lines. The JIB* module also provides signals to the RAM, EPROM, LSR, and CLOCK modules (V13, V). The P-GEN module provides signals to the C/S Interface module (3, 16) and the LSR module (V13, V). The C/S Interface module provides signals to the ABUS and AP lines. The RAM, EPROM, LSR, and CLOCK modules are connected to the V13 and V lines.

ORGANIZATION

IBM Internal Use Only

REGISTER DEFINITION

LOCAL STORAGE REGISTERS

The JIB' microprocessor is capable of addressing 128 Local Storage Registers (LSR's). These are contained in one SAMANTHA module. One class of instructions addresses registers as one byte fields, another class of instructions treats registers as even/odd pairs (halfwords). Therefore, the data path to and from the local storage registers and JIB' is organized as an 18 bit bus (16 bits of data and 2 bits of parity).

Register data out busses from JIB' (DO,DE BUS OUT) provide the outgoing data to the LSR's and External Registers (XR's). The DE BUS OUT provides the data path to the even addressed LSR's (LSB of register address is zero). DO BUS OUT provides the data path to the odd addressed LSR's and the XR's.

Incoming data from LSR's and XR's to JIB' is carried on data bus in (DE, DO BUS IN). The DE BUS IN provides the data path for the even addressed LSR's. The DO BUS IN provides the data path for odd addressed LSR's, and all XR's.

Addressing and data bus steering for LSR's and XR's is performed by the JIB' microprocessor module through appropriate address and control lines. Figure 14 shows the addressing structure for the LSR's and the relationship between the LSR field of a microinstruction and the LSPR (Local Storage Page Register).

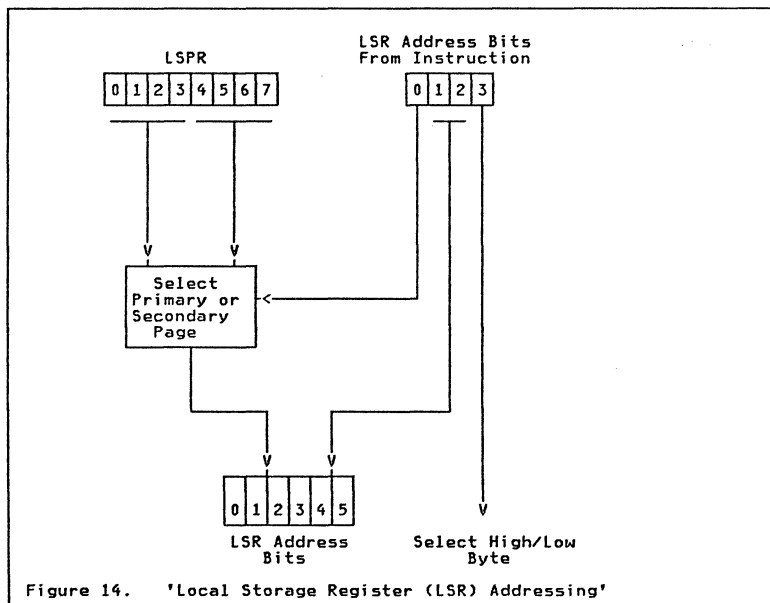


Figure 14. 'Local Storage Register (LSR) Addressing'

IBM Internal Use Only

EXTERNAL REGISTERS

EXTERNAL REGISTER GENERAL DESCRIPTION

The JIB' microprocessor provides addressing capability for up to 32 external registers (XR's), where each XR identifies a function or a collection of functions represented by up to 8 bits. One XR (address 31) is reserved for the Local Storage Page Register (LSPR) and is implemented within the JIB' microprocessor module. Additional XR's are implemented for control of the interrupt structure, the UBI module and error handling and diagnostics.

The JIB' transfers data to and from the XR's using the odd half of the busses used for the LSR's. Odd parity is maintained on these busses, but parity may be ignored if the Inhibit DE/DO Bus Check line to JIB' is held high. A bidirectional bus to and from the XR's is generated on the APSCA card to minimize the number of card and module pins required by the XR data busses.

Figure 15 on page 22 shows the address and function of all external registers implemented on the APSCA card.

EXTERNAL REGISTER PARITY CONTROL

Bit 0 of XR X'1C' controls parity checking on the XR data bus into JIB'. If this bit is B'1', the DE/DO check in JIB' is disabled. This bit is initially set to B'0'.

IBM Internal Use Only

XR Address	XR Name
1F	Local Store Page Register (LSPR)
1E	Interrupt Request Register
1D	Interrupt Mask Register
1C	XR Stack / C/S Page / HAZEL Interrupt and Inhibit DE/DO Check
1B	Trigger Address IPL flag and HAZEL Reset
13	External Register D
12	External Register C
11	External Register B
10	External Register A
0D	UBI Common Data Bus
0C	UBI Command Data
05	UBI Status B
04	UBI Block Address
03	UBI Control Register
02	UBI Data In
01	UBI Status A

Figure 15. 'External Register Assignment'

IBM Internal Use Only

COMMUNICATIONS

Communications are categorized as follows.

1. APSCA to Remote DIS
2. APSCA to Local I/O
3. APSCA to Hazel A
4. APSCA to APSCA (peer to peer)

APSCA/REMOTE DIS COMMUNICATIONS

General Operation

Communications between the RDIS units and the APSCA is carried out over a single coaxial line operating at a data rate of 1.25 Mbits/sec. The Attached Processor Source Communications Adapter (APSCA) card provides the function of interfacing Hazel A memory to the serial RDIS channel. As illustrated in Figure 1 on page 2, the APSCA connects to a maximum of 31 RDIS blocks.

Assuming that the APSCA has data ready to send to the RDIS, the data is transferred to the UBI module and the UBI module adds a CRC code and a start flag, and sends the string to the first RDIS block in the chain. At the RDIS block the message is de-serialized and decoded as well as re-shaped and sent upstream to the next block. In this way, all the RDIS blocks see the same message.

The block that the message was intended for responds with a return message to the APSCA. When data is traveling in this direction (back to the APSCA), the RDIS blocks between the block sending the message and the APSCA act as repeaters for the signal. The message reaches the APSCA where it is de-serialized and cycle stealed into Hazel 'A' memory.

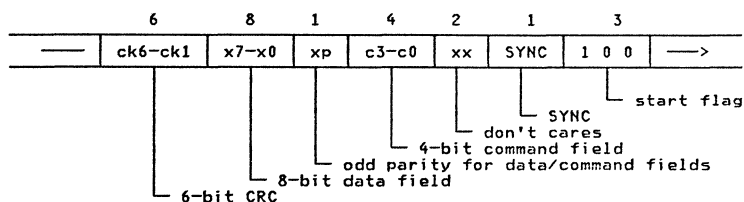
All transactions are initiated by the Hazel A processor and conform to the East Fishkill DIS Bus architecture. When Hazel A is not utilizing the communications line, the APSCA is polling all the blocks on the string for interrupts.

Transaction Format

All transactions between the APSCA and RDIS blocks are sent in small packs of 25 bits in length. Each pack includes a 3 bit start flag, a 7 bit control field, an 8 bit data field w/parity, and a 6 bit CRC code for the data and control fields.

The format for a message from the APSCA to a RDIS block is given below.

IBM Internal Use Only



IBM Internal Use Only

The Universal Block Interface (UBI) module is viewed by JIB' as a group of external registers. These registers are the means by which JIB' controls both the serial interface to the remote blocks and the local block interface. For more detailed information, see the Functional Specification on the Universal Block Interface (UBI) module.

Line Characteristics

The communications line has the following characteristics:

Maximum Distance Between Blocks	5,000 ft.
Data Rate	1.25 Mbits/sec
Data Code	PWM
Data Format	25 bit asynch.
Transmission line	93 ohm coax (RG-62)
Transmit Amplitude	8v peak-to-peak (Max)
Receiver Sensitivity	0.7v peak-to-peak (Min)
Line Delay	1.4 ns/ft
Repeater Delay	400 ns

APSCA TO LOCAL I/O COMMUNICATIONS

Communications to the local I/O is accomplished in the same manner as in the remote interface except that the parallel i/o interface of the UBI module is used. Local I/O is limited to 4 I/O cards which can be plugged into any of the 4 remaining slots of the Ariel box.

The APSCA distinguishes between local and remote blocks by the fact that the local block address is always 0. All other blocks from 1-31 will be remote blocks. This fact is needed by APSCA firmware when determining I/O Return In timeouts.

The parallel interface is comprised of a common 8 bit data bus, 4 command bits, a parity in bit, a parity out bit, 4 I/O select lines, 4 I/O interrupt lines and 2 handshake lines, Sync Out and Return In. These lines are directly connected to the ABUS connector from the UBI module. Local I/O timings are as shown in Figure 16 on page 26.

APSCA TO HAZEL A COMMUNICATIONS

Communications between the APSCA and the Hazel A are accomplished via a cycle steal interface. This interface is derived from the Jib' data and address busses. These busses are interfaced to the ABUS connector via open collector drivers for the data and tri-state drivers for the address. The data drivers are bi-directional and both these drivers and the address drivers are controlled by the Jib'.

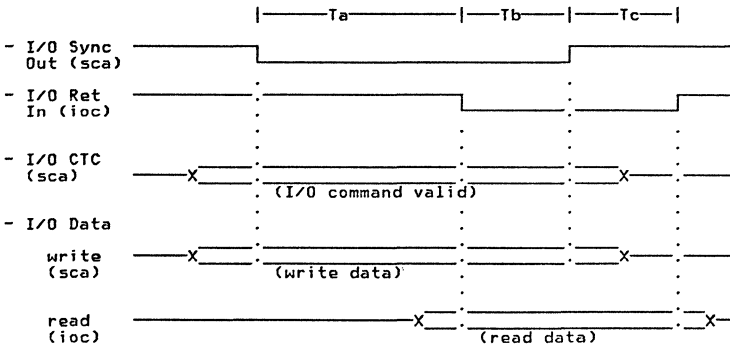
In addition, cs select and cs write are generated on the APSCA and interface to the ABUS via tri state drivers. These drivers are enabled by c/s acknowledge from Hazel A.

All cycle steal operations are controlled via the Jib' CS1H instruction. This opcode is detected by logic in the clock module which, in turn, generates a cycle steal request to Hazel-A.

Figure 17 on page 27 shows the timings for these interfaces.

IBM Internal Use Only

Local I/O Bus Control by APSCA



TIMING PARAMETERS

Symbol	Description	MIN us	TYP us	MAX us	see note
Ta	-I/O Return Response Time	1.5		30	1
Tb	-I/O Sync Out reset response time(write)	1.5		30	2
Tb	-I/O Sync Out reset response time(read)	3		30	2
Tc	-I/O Return in reset response time	.5		3	3

NOTE 1: The -I/O Return Response time is a function of the type of card being selected. Most cards will respond within 1 usec, however other cards which have on board micros can take as long as 30 usec to respond.

NOTE 2: -I/O Sync out will be reset by the APSCA firmware once -Return In has been detected. This time varies between read and write operations, and in the event of the occurrence of an error.

NOTE 3: Once -Sync is reset -return will be reset. This time is a function of the card being de-selected.

Figure 16. ARIEL Bus Local I/O Operations

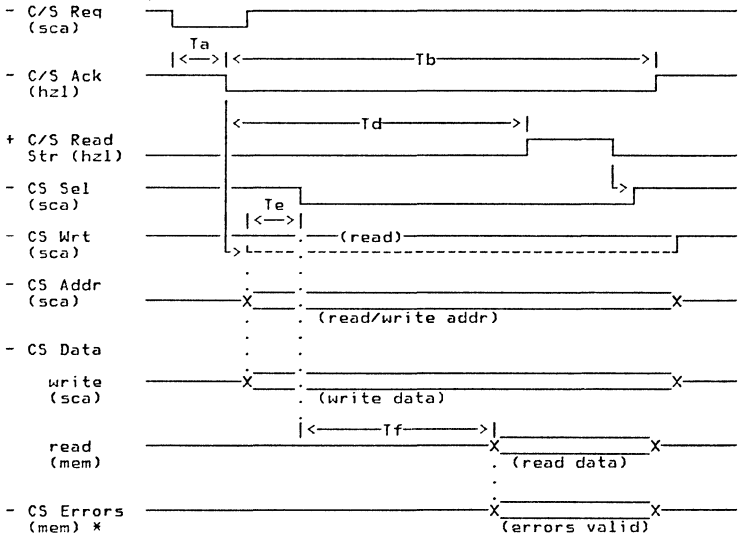
APSCA TO APSCA (PEER TO PEER) COMMUNICATIONS

This form of communications is intended for use in dual engine systems only and provides a high speed interface between 2 Hazel A engine cards thus providing peer to peer communications without going through the area controller.

All transactions will be initiated via the EDXJ user programmer through macros which will be defined at a later date. The ARIEL nucleus will require support code for this feature. The APSCA firmware will interpret the transfer commands and begin a set of I/O sequences using the local I/O interface. In this configuration there can be no local I/O cards and there will be 2 separate ABUS's. See Figure 18 on page 29

IBM Internal Use Only

Cycle Steal Acknowledge Detail



- * Control store errors include ECC single and double bit errors, address and data parity errors, key bit check, and write protect error.
 - (sca) - APSCA output
 - (hzl) - HAZEL-A engine card output
 - (mem) - memory card output

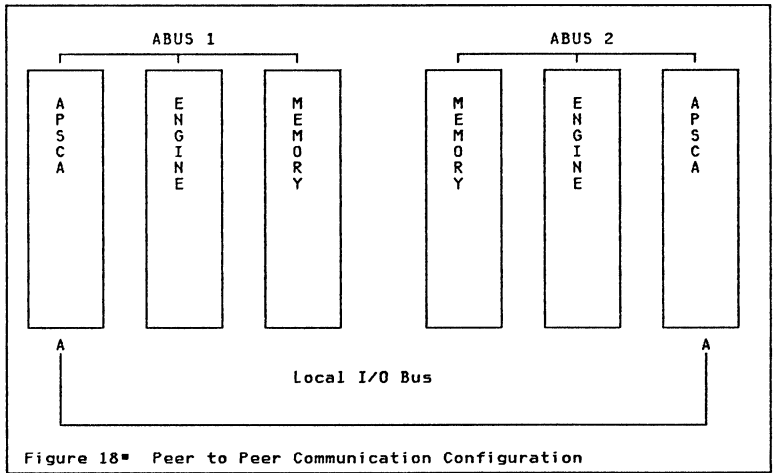
Figure 17. ARIEL Bus Cycle Steal Interface Timing

Communications is initiated when the APSCA puts the command and the data on the bus and activates I/O Sync. I/O Sync of one APSCA is wired to Interrupt 0 of the other APSCA so when I/O Sync is active the other APSCA is interrupted. The second APSCA responds to the interrupt by activating I/O Return Out which is wired to the interrupting APSCA's I/O Return In. This procedure is reversed when the second APSCA wishes to interrupt the first. see Figure 19 on page 30

IBM Internal Use Only

TIMING PARAMETERS for Figure 17

Symbol	Description	MIN ns	TYP ns	MAX ns	see note
Ta	C/S ACK response time	50		550	1
Tb	C/S ACK pulse width		417		
Tc	C/S REQ reset		10		2
Td	C/S ACK to C/S READ STR		292		
Te	addr/data and write line setup time		40		4
Tf	read data valid from CS SEL			200	5
Tg	C/S ACK off (continuous request)		583		2
NOTE 1: The C/S ACK response is a function of the system configuration. The time given is for a single active request and reflects the fastest possible response. When multiple requests are active, higher priority devices are acknowledged first. NOTE 2: The C/S REQ line is typically reset immediately after C/S Ack from hazel comes in. NOTE 4: The CS SEL line goes active 40 nsec after the CS WRT, CS ADDR, and CS DATA lines have settled. CS SEL is reset by the high to low transition of the C/S READ STR line. NOTE 5: All read data and error lines are valid from Tf to the end of CS SEL.					



IBM Internal Use Only

Peer to Peer Interface Timing

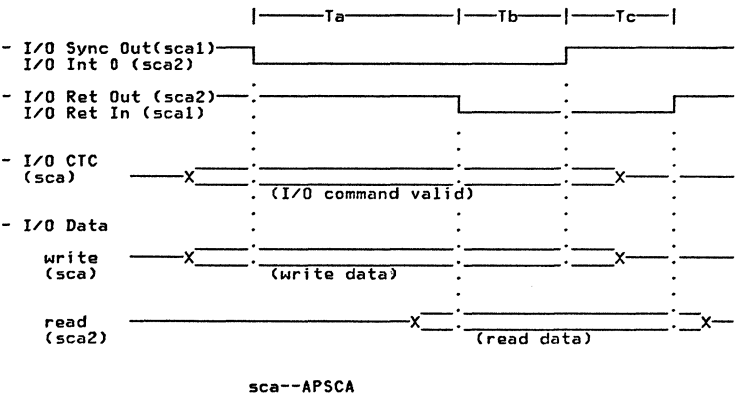


Figure 19. Peer to Peer Communications Timings: T_a , T_b , and T_c times are yet to be determined.

IBM Internal Use Only

CLOCK

The clock circuitry generates all the timing waveforms required by the JIB' microprocessor and Control Storage. It is driven by a 20 MHz crystal oscillator. Figure 20 on page 33 shows the waveforms generated.

The APSCA uses the same clock module as used in the old 'Hazel' system AP's. As a result of this there are many functions of the module which are not used.

JIB' TIMING

TA, TB, TC and TD are eight non-overlapping subcycle time periods required by the JIB' microprocessor. During cycle steal these waveforms are held down effectively stopping the microprocessor for one cycle while the external cycle steal device accesses memory.

This feature however, is not supported on the APSCA.

The hardware of the clock will ensure that the memory access timings are met by stretching JIB' T1 time. JIB' samples memory data at the end of T2. To fulfill this requirement, when instructions or data are being accessed from EPROM or RAM memory, one machine cycle time is 550 nSec. When the APSCA accesses HAZEL memory, T1 is stretched until the cycle steal acknowledge is dropped.

IBM Internal Use Only

MEMORY SELECTS

Two lines are used to select memory.

1. Sel Prom 2 & 3 --- since the RAM cycle time is the same as the EPROM this line is used to select RAM and is active if CS Select from Jib is active and the address bus is in the range of x'2000' to x'3FFF', however the 2k of RAM's address is valid only to x'27FF'.
2. Sel Prom 0 & 1 --- this line is used to select the EPROM and is active in the range x'0000' to x'1FFF'.

IBM Internal Use Only

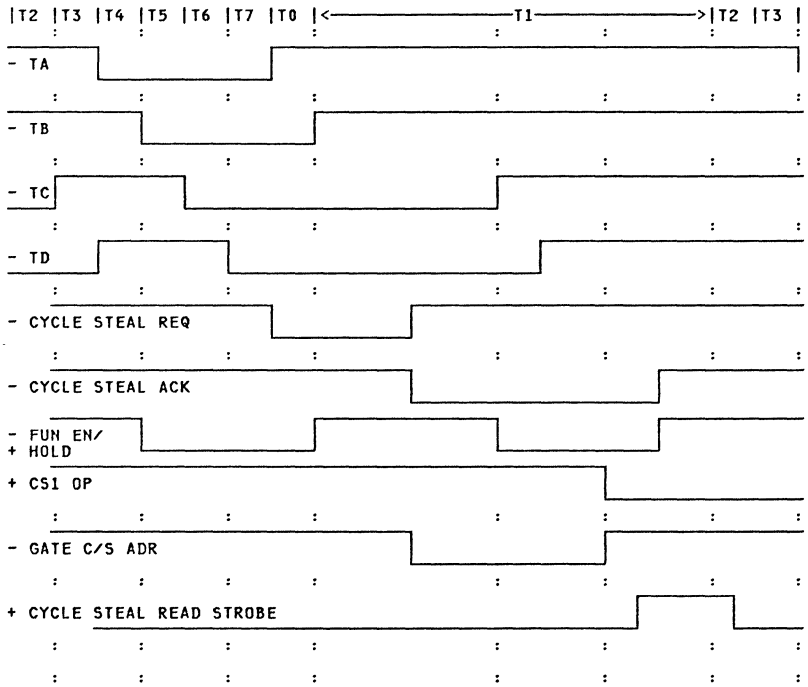


Figure 20. Clock Module Timings: CS1 Op will go active at T2 if CS Select is active.

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

EXTERNAL REGISTERS

EXTERNAL REGISTER INTERFACE

OVERVIEW

The external register interface is the primary interface for the APSCA to control the following:

- Communications
- Interrupt Handler
- Interface to Tool
- Error Handling
- Diagnostic Functions

ADDRESSING

The addressing of the XR's is provided via the Jib' XR Address Bus (5 bits plus parity) allowing up to 32 registers to be selected. The address lines are negative active.

DATA INTERFACE

The data interface is a 9 bit bidirectional tristate bus, negative active plus one parity bit. Checking of the parity bit can be inhibited via XR X'IC' bit 0.

The bus is used by all XR's with the exception of the ones contained in external register module pn.8494715, where a bus in and bus out structure is used.

CONTROL LINES

Control lines provide for selection, data gating, timing control, and directional control on the data bus. The timing is shown in Figure 21 on page 36.

XR SELECT: The SELECT line is negative active, and indicates that a valid address is present on the address lines.

SET REG GATE OUT: This line controls the gating of the tristate drivers. When the line is negative, data is transferred from the APSCA to the appropriate register.

REG LOAD COMMAND: This line controls the gating of data into the attached device. It is negative (i.e. active), only when SET REG GATE is negative. The attached device uses this line to gate data from the data bus into the XR's.

CLOCK HOLD: Not used

20 MHZ CLOCK DELAYED: Not used

XR DATA: These lines carry the data between the Jib' and the appropriate register.

IBM Internal Use Only

XR ADDRESS: These lines carry the XR address as contained in the JIB' instruction.

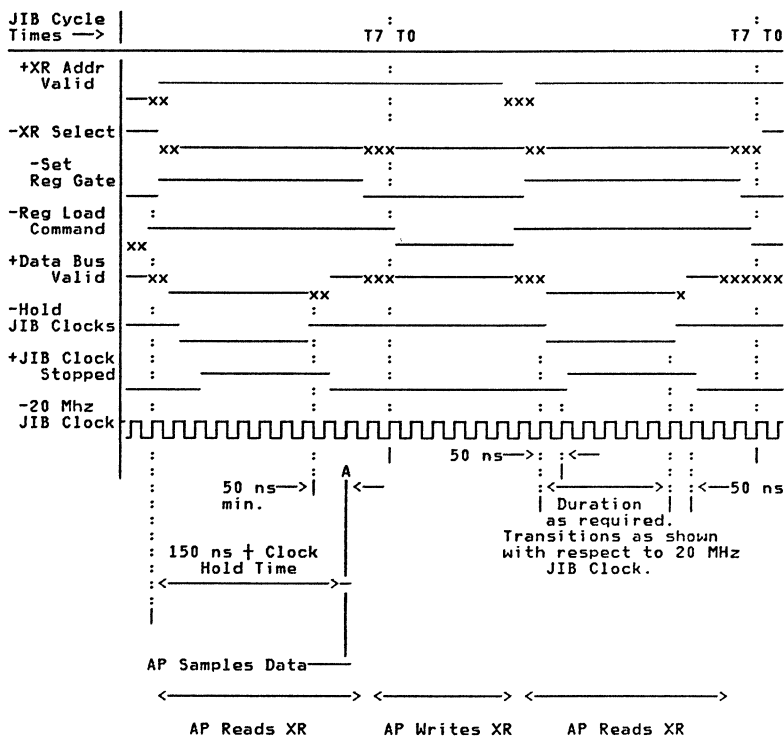


Figure 21. 'External Register Interface Timing': Note: The AP may cycle the XR's between Read and Write as fast as shown here. It is possible to have "dead" periods between read and write cycles as determined by the AP program.

IBM Internal Use Only

EXTERNAL REGISTER BIT ASSIGNMENTS

The following figures show all the external register bit assignments and bit definitions.

UBI Interface Registers

There are 7 registers which the local controller uses to program the UBI module:

#STAT A	: STATUS A,	hex'01', READ ONLY
#IND	: IN DATA,	hex'02', READ ONLY
#UCNTL	: UBI CONTROLS,	hex'03', READ/WRITE
#LBADDR	: LOCAL BLOCK ADDRESS,	hex'04', READ ONLY
#STAT B	: STATUS B,	hex'05', READ ONLY
#OUTC	: COMMAND OUT,	hex'0C', READ/WRITE
#OUTD	: DATA OUT,	hex'0D', READ/WRITE

These registers provide the status and control lines enabling the local controller to interface to the host processor and the macro-function cards when in modes 0 and 1. Access to these registers is through the MPI/O data bus and the XR address bus.

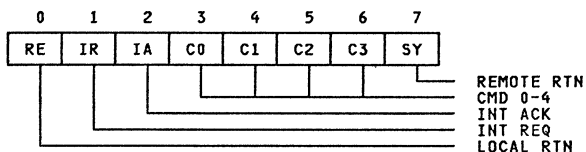
What follows is a description of each register. Symbol N/A is used with bit description to indicate that a particular bit is not used in MODE 0. The value shown for each bit is the active state, unless stated otherwise.

IBM Internal Use Only

NAME: #STAT - STATUS A VALUE: hex'01' READ ONLY

If all interrupts (INT0, INT1 and INT2) are tied up to the same level in AP, then bits 0, 1 and 7 may be used to determine the reason of an interrupt from the UBI module.

When INT ACK or INT REQ signals are active, the LOCAL RTN and REMOTE RTN bits should be used to determine wheather these signals originate from a local or a remote block.



- Bit 0 - LOCAL RTN '1' - Return from local macrofunction cards is activated. INT1 is generated.
- Bit 1 - INT REQ '1' - Interrupt request from BIC is active (either from a local or a remote block). INT2 is generated.
- Bit 2 - INT ACK '1' - Interrupt acknowledge from BIC is active (either from local or a remote block) Valid only during a CTC0.
- Bit 3-6 - CMD 0-4 '0' to 'F' - Current value of CMD bus.
- Bit 7 - REMOTE RTN '1' - Return from remote macrofunction cards is activated. INT0 is generated.

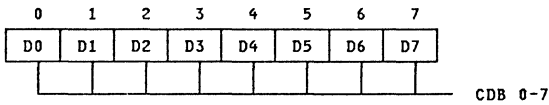
IBM Internal Use Only

NAME: #IND - IN DATA VALUE: hex'02' READ ONLY

This register is used to read data from the macrofunction cards during a DIS reads from both local or remote blocks.

Note, that X'00' must be written to #OUTD register before reading #IND register.

In accordance with the DIS read protocol #IND may be read when RETURN signal from a macrofunction is present. In this application data is ready after the local or remote return is active.



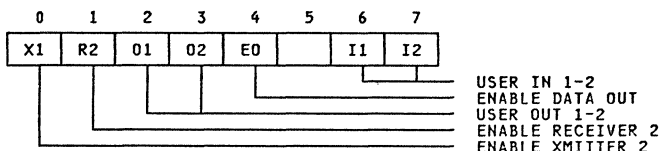
Bit 0-7 - CDB 0-7 'X' - Current value of Common Data Bus.

IBM Internal Use Only

NAME: #UCNTL - UBI CONTROL VALUE: hex'03' READ/WRITE

Used to select different modes of operation. The bits in this register have no effect if the MPC control pin is inactive (+5v).

Since port 1 will be used in this application, bits 0 and 1 should be initialized to '0'. The programmer should be careful not to alter accidentally bits 0 and 1 when changing other bits in this register.



- Bit 0 - ENABLE XMITTER 2** '0' - Transmit data is sent to xmit port 1. (POR state)
 '1' - Transmit data is sent to xmit port 2. (wrap test)
- Bit 1 - ENABLE RECEIVER 2** '0' - Data received over port 2 is ignored. (POR state)
 '1' - Data received over port 2 is decoded and loaded into RCV Buffer. (for wrap test)
- Bit 2 - USER OUT 1** 'X' - User programmable output. Reserved for reset of INT0. This bit must be toggled 0->1->0 to accomplish the reset operation.
- Bit 3 - USER OUT 2** 'X' - User programmable output. Allows control of external hardware (e.g LEDs). Available if BKC is low.
- Bit 4 - ENABLE OUTPUT** '0' - LOCAL block I/O. Drivers for CMD and CDB busses are put in high impedance state. (POR state)
 '1' - LOCAL block I/O. Contents of #OUTD and #OUTD are output to CMD and CDB respectively.
 In general, this bit switches registers #IND, #OUTD, #OUTC #stata(1-6) between LOCAL and REMOTE blocks.
- Bit 6-7 - USER IN 1-2** 'X' - User test inputs. Allows attachment to external input hardware (e.g. switches).

IBM Internal Use Only

NAME: #LBADDR - LOCAL BLOCK ADDRESS VALUE: hex'04' READ ONLY

It contains the block address and the current LSA of the local block. No corresponding register exists for the remote blocks.

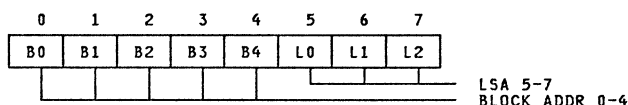
The block address is read directly from the program pins and the LSA from the LSA register on the BIC. To change the LSA, the user must issue a CTC0 to the BIC. This register has two formats depending on the value of BKC.

When polling for interrupts it is essential to determine if the currently polled block is local or remote. The address of the local block can be determined by reading the #LBADDR register (bits 0-4) or (bits 0-3) - depending on the configuration.

The configuration of the remote blocks can be set using the Block Configuration program pin (BKC).

Note: in this application the remote block will always be wired to have address x'00'.

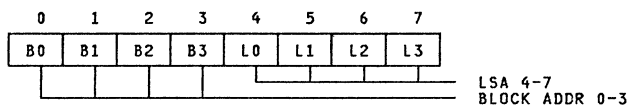
1. BKC = low (gnd): 32 by 8.



Bit 0-4 - BLOCK ADDR 0-4 'X' - 5-bit Block Address. (positive logic)

Bit 5-7 - LSA 5-7 'X' - 3-bit Logical Space Address. (positive logic)

2. BKC = high (+5v): 16 by 16.



Bit 0-3 - BLOCK ADDR 0-3 'X' - 4-bit Block Address. (positive logic)

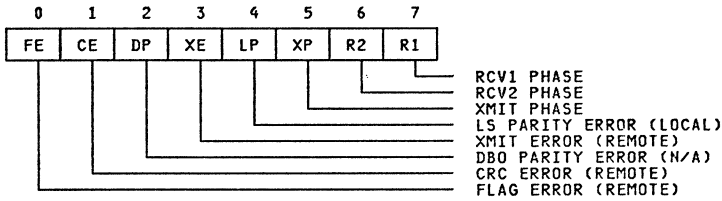
Bit 4-7 - LSA 4-7 'X' - 4-bit Logical Space Address. (positive logic)

IBM Internal Use Only

NAME: #STATB - STATUS B VALUE: hex'05' READ ONLY

Bit 6 should be used only in bringups.

Bits 7 may be checked after INT0. Bit 5 may be checked just before loading #OUTD or #OUTC. If these bits are not '0' one should wait until they are '0' before starting any I/O. Also, this condition should be logged as a H/W error resulting from a noise on the bus.



- | | |
|---------------------------------|---|
| Bit 0 - FLAG ERROR | '1' - Flag error indicator in serial mode. Valid while INT0 is active. Remote block. |
| Bit 1 - CRC ERROR | '1' - CRC error indicator in serial mode. Valid while INT0 is active. Remote block. |
| Bit 2 - DBO PARITY ERROR | '1' - Parity error indicator for data and command tag received from host. Valid while INT0 is active. N/A |
| Bit 3 - XMIT ERROR | '1' - Logical OR of bits 0 and 1. |
| Bit 4 - LS PARITY ERROR | '1' - Parity error indicator for data received from macrofunction cards. Valid while INT1 is active. Local block. |
| Bit 5 - XMIT PHASE | '1' - Transmit Phase indicator. |
| Bit 6 - RCV2 PHASE | '0' - Port 2 not active.
'1' - Port 2 receiving data. |
| Bit 7 - RCV1 PHASE | '0' - Port 1 not active.
'1' - Port 1 receiving data. |

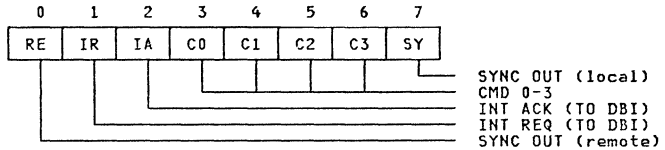
IBM Internal Use Only

NAME: #OUTC - COMMAND OUT VALUE: hex'0C' READ/WRITE

Used to write data to CMD bus and toggle the SYNC line.

Bits 1, 2 should be set to '0' during the initialization and never accidentally set to '1'.

The user must ensure that both bits 0 and 7 are not '1' at the same time; the local and remote SYNC line should not be high simultaneously.

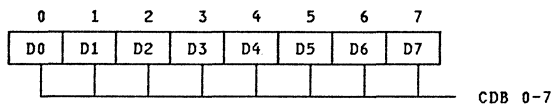


- Bit 0 - SYNC REMOTE '1' - Rises sync to REMOTE block. Activates Return line back to the host.
- Bit 1 - INT REQ '1' - Activates Int Req line back to the host. Keep '0'.
- Bit 2 - INT ACK '1' - Activates Int Ack line back to the host. Keep '0'.
- Bit 3-6 - CMD 0-3 'X' - These four bits are output to the CMD bus if bit 4 of #UCNTL is high.
- Bit 7 - SYNC LOCAL '1' - Activates SYNC line to the LOCAL macro-function cards.

IBM Internal Use Only

NAME: #OUTD - DATA OUT **VALUE:** hex'0D' **READ/WRITE**

Used to write data to the Common Data Bus.



Bit 0-7 - CDB 0-7 'X' - These eight bits are output to the Common Data bus if bit 4 of #UCNTL is high.

IBM Internal Use Only

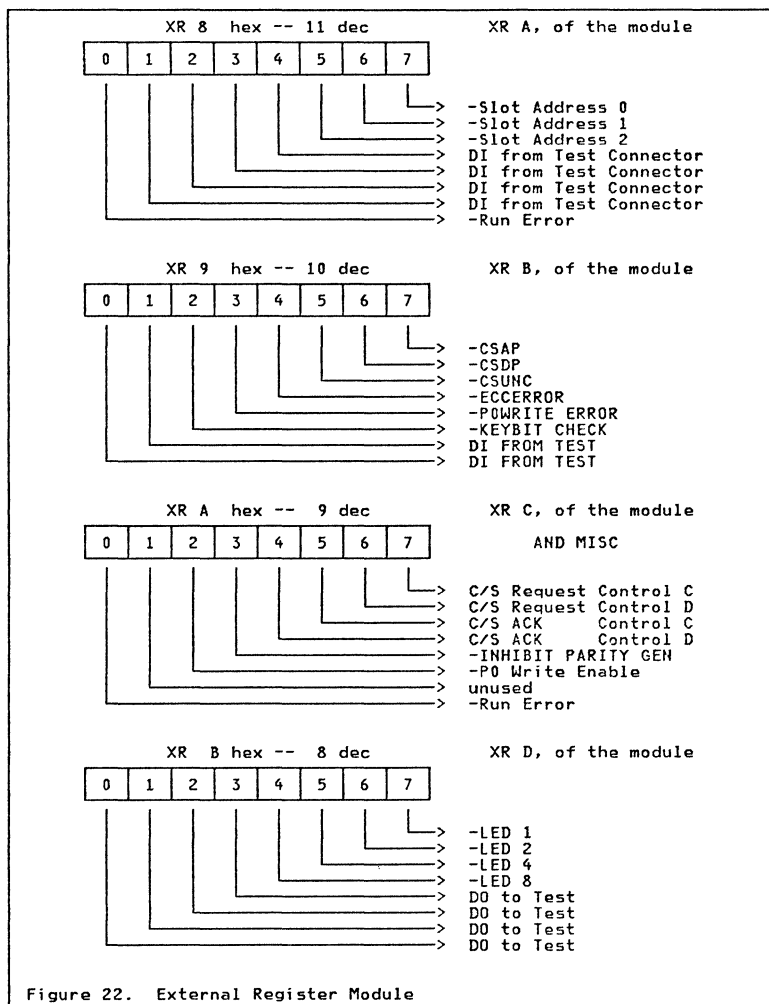


Figure 22. External Register Module

IBM Internal Use Only

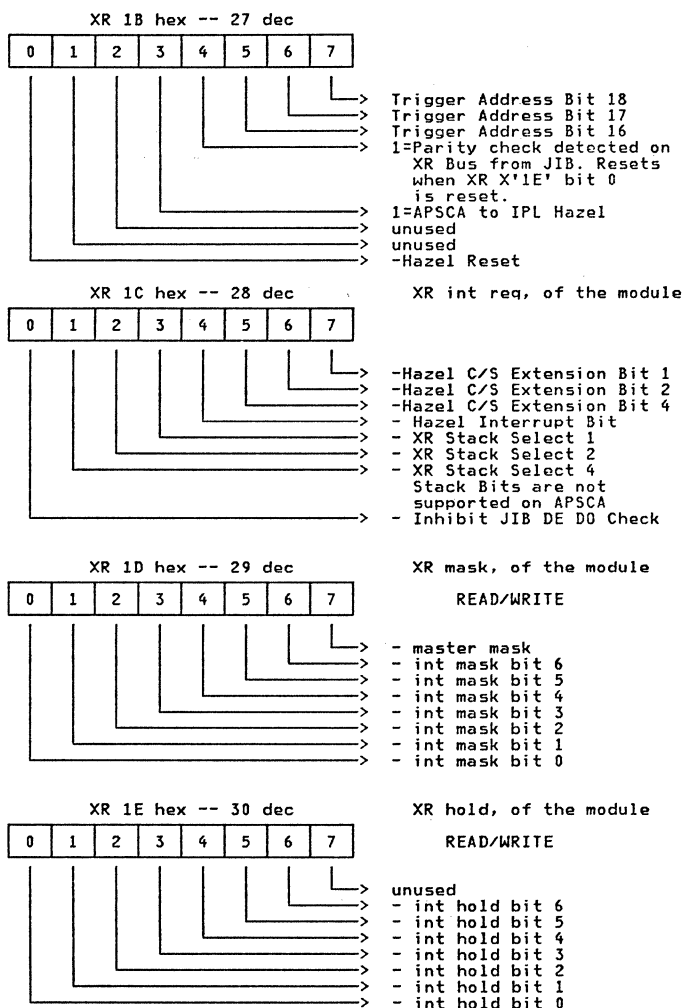
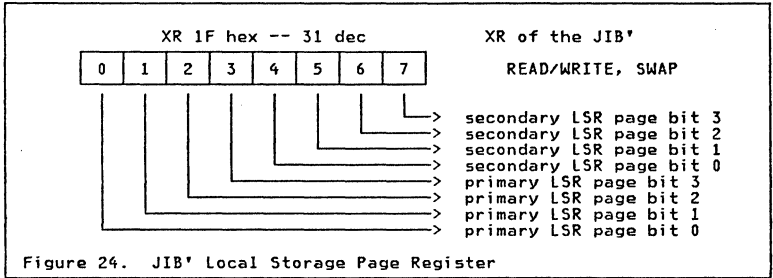


Figure 23. XR's of the Interrupt Module U13

IBM Internal Use Only



IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

INTERRUPT STRUCTURE

GENERAL

APSCA interrupt facilities include the following capabilities:

- 8 Prioritized interrupt levels.
- Automatic hardware controlled PSW swap.
- PSW storage are for inactive levels stored in LSR's
- Automatic latching of external interrupts

The interrupt input to the Jib' module comes from the interrupt handler module. The interrupt module will support up to 8 interrupt input lines and for APSCA normal operations four of these lines will be used. Of the four remaining, three can be used for diagnostics and testing and one is always active. The following are the interrupt level assignments:

LEVEL	FUNCTION
0	Machine Check
1	Hazel Cycle Steel Address Compare
2	Interrupt from test point
3	-Interrupt 2 from UBI Module
4	-DBO Sync from UBI Module
5	Interrupt from test point
6	Interrupt from test point
7	Always active
	Level 0 is the highest priority level.

Figure 25. 'Interrupt Level Assignments'

Interrupt Level Definitions

- Level 0** An interrupt on this level causes the Machine Check Handler to be invoked. The line is activated when the -Jib Error line from the Jib' is active or during a cycle steal operation if -CS Any Error goes active.
- Level 1** This line is generated internally in the Interrupt module and becomes active when -AP Trigger (from Hazel-A) is active and the trigger address (from Hazel A address) matches the slot address. Used in APSCA to Hazel-A Communications.
- Level 2,5,6** These interrupt lines come from the test connectors located at the top of the card. They can be used for test and diagnostics purposes.
- Level 3** This is the Interrupt 2 line from the UBI module and represents one of four possible interrupts from the local I/O cards. It is active any time the reference register does not match the interrupt register of the

IBM Internal Use Only

UBI module. The local I/O blocks are distinguished from the remote blocks by the facts that the local block will interrupt on this level and the block address for a local block is always zero.

- Level 4** This is the DBO Sync line from the UBI modules and becomes active any time data has been returned from one of the remote blocks. The firmware uses this interrupt to interpret when a response from a remote block is received. An interrupt from a remote block is detected when the interrupt request bit is active in UBI external register x'0C'.
- Level 7** This level is always active and can be used for 'idle' routines.

APSCA INTERRUPTS TO HAZEL A

The APSCA uses XR x'1c' bit 4 to interrupt the Hazel A. This bit is multiplexed to either interrupt request 0,2,3 or 4 on the ABUS. The interrupt line used is controlled via XR x'12' bits 4 and 5 which are to be set during APSCA bringups.

THEORY OF OPERATION

JIB' samples its interrupt request line at the T7 time of the JIB' cycle. If the line is active, microinstruction fetch and execution is suspended and the microprocessor automatically enters a sequence to store the program status word (PSW) of the currently executing interrupt level into a predefined area within the LSR's. The PSW of the level corresponding to the next interrupt level is swapped out of its LSR's into JIB' and instruction execution proceeds in the new level, using the new PSW. The next interrupt level is passed to JIB' on 3 encoded lines. This requires that the interrupt hardware external to JIB' prioritize and encode the new interrupt level.

Some JIB' instructions inhibit the sampling of interrupts. XR instructions reading or writing XR addresses from X'24' to X'31' inclusive cause interrupt sample to be inhibited. Also, interrupt sample is inhibited during the first cycle of all two cycle JIB' instructions (CS0, CS1, and some LSR to LSR instructions). In these cases, the interrupt will be sampled at the end of the next cycle (unless also inhibited by this mechanism).

The program status is four bytes of information consisting of the Instruction Address Register (IAR), Local Storage Page Register (LSPR), the condition codes (CCR) and error information (ERR). This PSW is stored into the area of LSR space assigned for that executing level. Figure 26 on page 51 shows the PSW format and assignment within LSR space. When the current PSW has been saved, the hardware automatically fetches the PSW from the LSR space assigned to the interrupt level about to resume execution. After the new interrupt level's IAR, LSPR and CCR have been restored (ERR is not restored; it is cleared to 0), the PSW swap is complete and normal microinstruction execution proceeds at the new level. The PSW swap time, i.e., the time delay between execution of the last microinstruction at the old level until the fetch of the first microinstruction at the new level, is 2 machine cycles or 1.0 microseconds.

Interrupts may be generated by the external hardware (an external interrupt), or by the microcode (a programmed interrupt). The facilities for doing these functions are described in the section on interrupt hardware.

IBM Internal Use Only

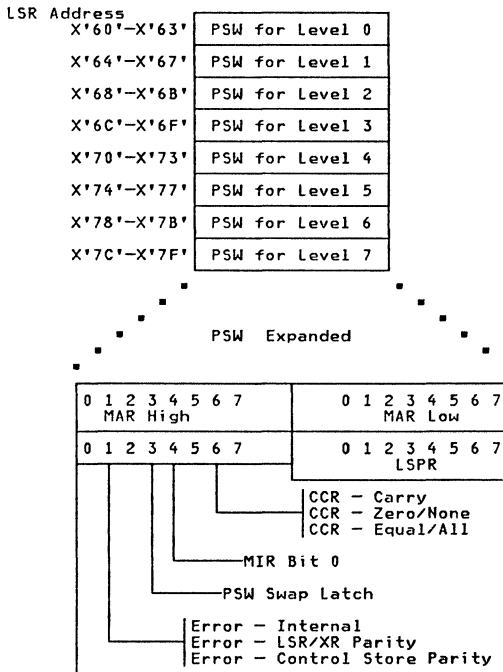


Figure 26. 'Program Status Word (PSW) Format'

INTERRUPT HARDWARE DESCRIPTION

OVERVIEW

The interrupt hardware provides the masking, priority and interrupt request facilities for 8 prioritized interrupts. The lowest priority interrupt is assumed to be always active and always masked on, leaving 7 interrupts available to JIB' and other hardware devices. When one of the 7 interrupt requests becomes active and is not masked off and is on a higher level than the currently executing level, the following sequence occurs:

The priority network generates a three bit code corresponding to the highest outstanding interrupt request level. The compare network, recognizing a mismatch between the encoded new level and current level, generates a level swap request to the JIB' microprocessor. The microprocessor will use its Current Interrupt Level register to generate the address to store the current PSW, and will use the interrupt level lines to generate the address to fetch the new PSW. At the completion of the PSW swap the microprocessor will set its Current Interrupt Level register from the interrupt level lines and will raise Interrupt Response. This resets the Interrupt Request

IBM Internal Use Only

Latch in the external hardware. One instruction in the new interrupt level is guaranteed to be executed before another interrupt level switch can occur, even if a higher level interrupt becomes active.

The interrupt hardware sets the bit corresponding to an external interrupt in the Interrupt Register. The register will hold this interrupt level active until it is reset by the JIB' program even though the original interrupt request line is deactivated. The last instruction of the interrupt service routine must turn off the corresponding bit for its level in the Interrupt Register. This will cause a PSW swap back to the next highest active level. Note that because resetting the bit cannot be immediately followed by an interrupt, and because of the external hardware JIB' will execute one or two more instructions (dependant on other external interrupts) before swapping levels. The level swap will occur even if the external interrupt request for the current level remains active. The external interrupt request will, however, not be lost.

Programmers should note that resetting a bit in the Interrupt Register other than the one corresponding to the current interrupt level may cause that interrupt to be lost. This will occur if the external interrupt is no longer active.

To prevent problems within the mask and interrupt registers, these registers are assigned to external register addresses within the group which inhibits detection of interrupts. This enables the program to manipulate these registers without causing invalid or erroneous interrupts.

INTERRUPT MODULE

The Interrupt handler is implemented on one GOLF Module (Figure 27 on page 54) which includes:

The Interrupt Request Register

This register indicates the levels which have outstanding interrupts. A bit in this register containing B'1' indicates that an interrupt is pending on the corresponding level. The bit assignments are shown in Figure 23 on page 46. Bits may be set by either JIB commands to XR X'1E', or by external interrupts. Bits may be reset only by JIB commands.

Bits in this register may only be set by external interrupts if no interrupt request to JIB' is outstanding. External interrupts are sampled at JIB' T6 time and latched at JIB' T2 time, so that even if the external interrupt drops, the interrupt bit corresponding to that interrupt will remain on.

Note that bit 7 of this register is not implemented, and will always be B'0'.

POR resets this register to X'80' (Level 0 interrupt), forcing the next level lines to X'0', thereby ensuring the JIB' starts execution in level 0.

The Interrupt Mask Register

This register is used to mask the contents of the Interrupt Request Register. Only those interrupt levels which have the corresponding bits set to B'1' in this register may cause an interrupt. Bit 7 of this register is not used to mask interrupt level 7 (as this level is always active), but is used as a 'Master Mask'. If bit 7 is set to B'0' no interrupts will be generated to JIB. JIB' may set or reset

IBM Internal Use Only

this register via any XR instruction to XR X'1D'. Bit assignments for this register are shown in Figure 23 on page 46.

POR resets this register to X'80', allowing the level 0 interrupt to appear on the next level lines.

The Interrupt Priority Tree

This circuit determines the highest outstanding interrupt which has not been masked off. When this value is different from the Next Level Register, an interrupt is generated to JIB, and the Next Level Register is updated.

The Next Level Register

This register holds the next interrupt level on which JIB' is to operate. The register is not accessible from JIB' microcode, and is updated by the hardware at JIB' T2 time. This is required because the next level to JIB' must be stable before T7 time. POR Resets this register to X'0'. If the Interrupt Request Latch is on, updating the Next Level Register is delayed until the Interrupt Request is dropped.

The Level Comparator

This circuit compares the contents of the Next Level Register and the output of the Interrupt Priority Tree. If the two values are unequal, the Interrupt Request Latch is set at JIB' T2 time, simultaneously with the Next Level Register being updated.

Interrupt Timing

Setting or resetting a bit in either the Interrupt Request Register or the Interrupt Mask Register (other than the Master Mask) from microcode will take effect (causing a PSW swap), at the end of the second cycle following the instruction which caused the set/reset. This is caused by the internal clocking of the Interrupt GOLF module. This implies that the next instruction will always be executed and that the instruction following that may execute before the PSW swap occurs.

Setting or resetting the Master Mask will take effect during the instruction which set/reset the Master Mask. Note that one more instruction will always be executed in the current level before a PSW swap (if any) can occur because JIB' does not sample the interrupt request during the instruction which manipulates the Master Mask.

When an external interrupt goes active, JIB' will not swap PSW's for at least 2 cycles, since it takes this many cycles to propagate the interrupt through to the Next Level register and the Interrupt Latch.

IBM Internal Use Only

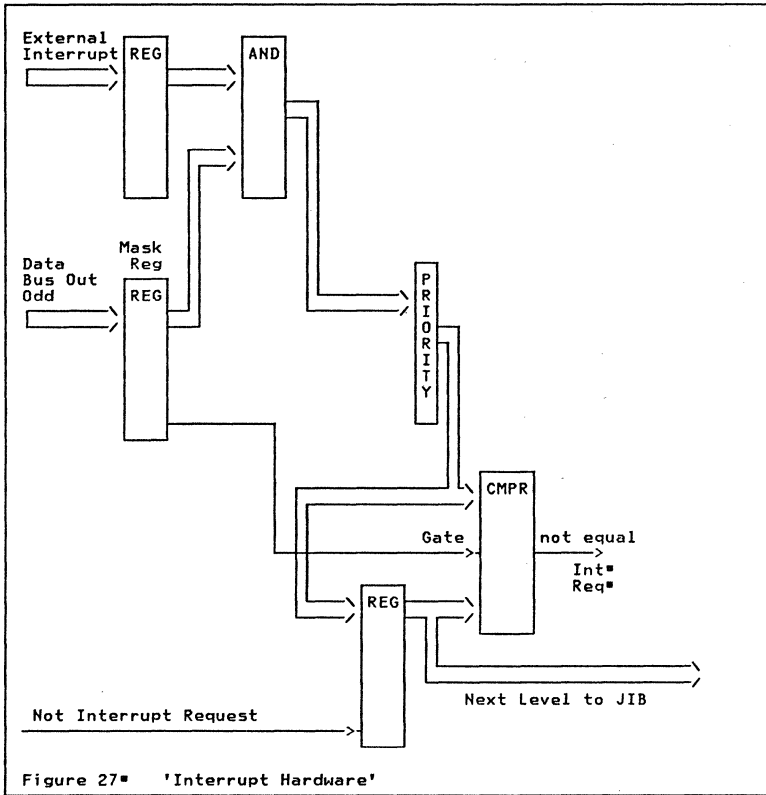


Figure 27# 'Interrupt Hardware'

IBM Internal Use Only

ERROR CHECKING

GENERAL

On the occurrence of a check condition detected by the microprocessor, the check is latched within JIB'. At the time of occurrence of a check the ERROR output line is raised by JIB'. This line is hard wired on the APSCA card to cause interrupt level 0 request with a subsequent PSW swap to that level. The error handling microroutine on level 0 (highest priority level) can then examine the error conditions stored in the PSW area of the LSR's for the level causing the error. The error condition bits are not restored to JIB' when the PSW is returned for the continuation of the microprogram on the error level.

Note that the PSW swap to level 0 is gated by the state of the interrupt mask bits. The PSW swap will not occur if either the master mask is off or if bit 0 of the interrupt mask is off.

JIB' DETECTED CHECKS

DE/DO Bus In Check: This check occurs whenever the DE or DO BUS IN does not have odd parity across either bus. The bus in is checked on all microinstructions during data load time, (i.e., anytime the busses are to contain correct parity from registers.)

If devices on the external register interface do not generate parity, they must raise Inhibit DE/DO check when selected to prevent this error from being detected. JIB' will then generate good parity for the input bus.

Control Store Data Check: This check condition occurs whenever bad parity is detected on the DE/DO bus in during a Control Store fetch of instructions or data.

JIB' Internal Check: This check condition occurs when bad parity is detected as a result of an ALU operation or whenever bad parity is detected in the MIR register.

ERRORS DURING A CYCLE STEAL OPERATION

-CS Any Error: This condition will occur during a cycle steal read or write from Hazel A memory. It is active any time any of the following signals is active.

- -Control Store Address Parity Error
- -Control Store Data Parity Error
- -Control Store Uncorrectable Error
- -Partition 0 Write Error
- -Keybit Check

These conditions are readable via external register X'11'. The -CS Any Error sets a latch on the APSCA, this latch is reset when external register X'11' is read.

IBM Internal Use Only

MACHINE CHECK HANDLER REQUIREMENTS

The code in the machine check handler is required to do the following if an error has occurred.

1. Dump local and external registers to Hazel A memory
2. Activate the -Run Error bit in XR X'10' bit 0. This will cause -run error to be active on the ABUS and cause the -Run Error light to come on.

IBM Internal Use Only

DIAGNOSTICS

GENERAL

The APSCA microprogram must include bringup diagnostics executed immediately after being powered on. Since the APSCA has no direct external interface, diagnostic results will be reported via four LED's mounted on the card. The LED's are to be read as a hexadecimal number.

DIAGNOSTIC COVERAGE

The LED display is reset to 0 (all lights off) on POR. The value displayed will be changed by the diagnostic microcode as follows:

- The LED is set to 1 by the first instruction to indicate successful JIB' startup.
- The LED is set to 2 after successful completion of LSR tests.
- The LED is set to 3 after successful completion of RAM tests.
- The LED is set to 4 after successful completion of interrupt hardware tests.
- The LED is set to 5 after successful completion of tests of XR space.
- LED values from 6 to A are reserved.
- LED values from B to F may be used by the application to indicate program states, at the user's option.

DIAGNOSTIC LED CONTROL

The LED's are controlled by XR X'13'. The bit assignments are shown in Figure 22 on page 45. Bits 4 to 7 contain the value to be displayed on the LED.

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

MICROINSTRUCTION SET

MICROINSTRUCTION DESCRIPTION CONVENTIONS

Field bit assignments are numbered from left to right beginning with 0. Arithmetically, the most significant bit is bit position 0, while the least significant bit is in position 7 or 15 in 1 or 2 byte fields respectively.

EPROM PROGRAMMING

The address lines to Control Store on the Attached Processor are active low (a low level is equivalent to an address bit value of B'1'), as are the Control Store data lines. This fact must be noted when writing the Intel 2764 EPROM's which normally will be used to hold the firmware program.

MICROINSTRUCTION SET DESCRIPTION

All microinstructions are fixed length of 16 bits plus 2 parity bits for checking. The microinstruction set is divided into 12 basic instructions as shown in Figure 28 on page 60.

Local Store Registers (LSR) are addressed by a 4-bit field in the microinstruction. The most significant bit of this field is used to select either the primary (high order) or secondary (low order) page field of the local store page register (LSPR). Thus, local storage is arranged in 16 pages of 8 registers per page.

Three condition code bits are used to indicate execution results of instructions. The setting of these bits is included with each instruction definition which follows.

IBM Internal Use Only

INSTR	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	CY	NOTES		
BR	0	0	0	0	BA												1	BA → MAR		
BOB-XR	0	0	0	1	BIT		XR		DISP								1	MAR+DISP→MAR,Conditionally		
XI	0	0	0	1	ALU		XR		K/LSR								1 or 2	ALU Ops 0 - AND 'KF' 8 - AND 'FK' 1 - OR 'K0' 9 - OR '0K' 2 - XOR 'K0' A - XOR '0K' 3 - ADD 'K0' B - ADD '0K' 4 - TUM 'K0' C - TUM '0K' 5 - LD 'K0' D - LD '0K' 6 - XR→LSR E - CMPR LSR/XR 7 - LSR→XR F - SHFT RT LSR		
	MAR + 1 → MAR XR,ALU,K → XR XR → LSR LSR → XR LSR / 2 → LSR CMPR takes 2 Cycles																			
BC	0	1	0	0	COND		1	BA									1	Branch Conditions 0 - Zero 4 - Equal 1 - Nonzero 5 - Unequal 2 - Carry 6 - Reserved 3 - No Carry 7 - Reserved		
	BA → MAR, Conditionally																			
CS0	0	1	0	0	0	I	F	0	CS	ADDRESS LOW							2	CS Addr=LSR 2 Address Low Data LSR= LSR 0,1 F=Fetch : I=Incr LSR 2,3		
CS1	0	1	0	0	1	I	F	0	LSR2	0	LSR1	0						2	CS Address = LSR2 Even/Odd CS Data = LSR1 Even/Odd F=Fetch : I=Incr LSR2	
CS1H	0	1	0	0	1	I	F	0	LSR2	0	LSR1	1								
BALR	0	1	0	0	1	1	0	0	LSR2	1	LSR1	0	1						1	LSR2 → MAR, MAR+1 → LSR1
COPY	0	1	0	0	1	1	1	0	LSR2	0	LSR1	X	1						1	LSR2 → LSR1 Even/Odd
BOBLSR	0	1	0	1	BIT		P	LSR		DISP							1	MAR+DISP→MAR,Conditionally		
BAL	0	1	1	0	BA												1	BA→MAR : MAR+1→LSR 4,5		
RR	0	1	1	1	ALU		0	LSR2		LSR1							2 or 1	ALU Ops 0 - AND 4 - TUM 1 - OR 5 - MOVE 2 - XOR 6 - COMPARE 3 - ADD 7 - ADD W/CARRY		
	MOVE takes 1 Cycle																			
RI	1	K	ALU		K	LSR		K								1				

Figure 28• 'Instruction Set'

IBM Internal Use Only

FIRMWARE SPECIFICATION

INTRODUCTION

This document is a high-level description of the AP-DIS firmware architecture.

HIGH LEVEL DESCRIPTION.

The firmware runs on five different levels, three of which are interruptible:

- LEVEL 0:
 1. M/C handler (h/w interrupt due to hardware errors)
 2. register dump (firmware interrupts)
- LEVEL 1:
 1. Debug requests (interrupt from Hazel)
 2. New work (interrupt from Hazel)
- LEVEL 3:
 1. Local block interrupt request (interrupt from UBI)
- LEVEL 4:
 1. Idle loop
 2. DIS I/O processor
 3. DIS command processor
 4. Remote interrupt poller and handler
 5. Local interrupt handler
 6. New work finished acknowledgement to Hazel
 7. Error log
- LEVEL 7:
 1. AP bringups and diagnostics
 2. UBI bringups and diagnostics
 3. firmware initialization

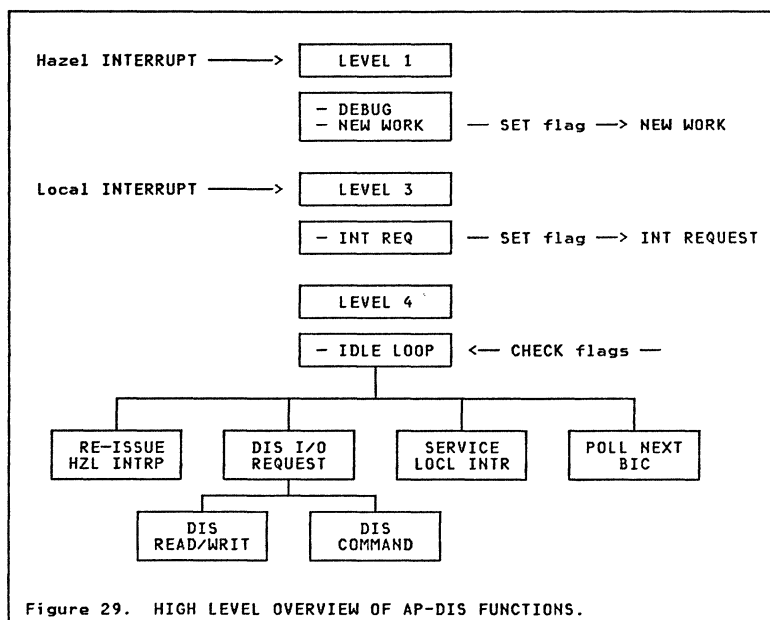
Levels 2, 3 and 6 are not used.

The only term that needs defining is 'new work'. New work is defined as a request from Hazel to AP to perform some work. Hazel interrupts AP on level 1 to get AP's attention. AP finds the definition of the new work in a control block called a DCB (Data Control Block). The DCB contains all information required by the AP to perform new work. After the new work is finished AP sets a return code in Hazel memory and interrupts Hazel on level 3 to signal that the new work was completed. New work can be either a DIS I/O request or a DIS command (e.g. UPDATE POLL LIST).

ARCHITECTURE: LEVEL1/LEVEL3/LEVEL4

Levels 1, 3 and 4 are the most important because all the DIS functions and debug are performed by the microcode running on these levels. Level 1 is a source of new work. Level 3 detects interrupts from the macrofunction cards in the local block. Level 4 carries out new work requests and then it responds to Hazel to signal completion of the new work. Level 4 also handles macrofunction card interrupts and logs errors.

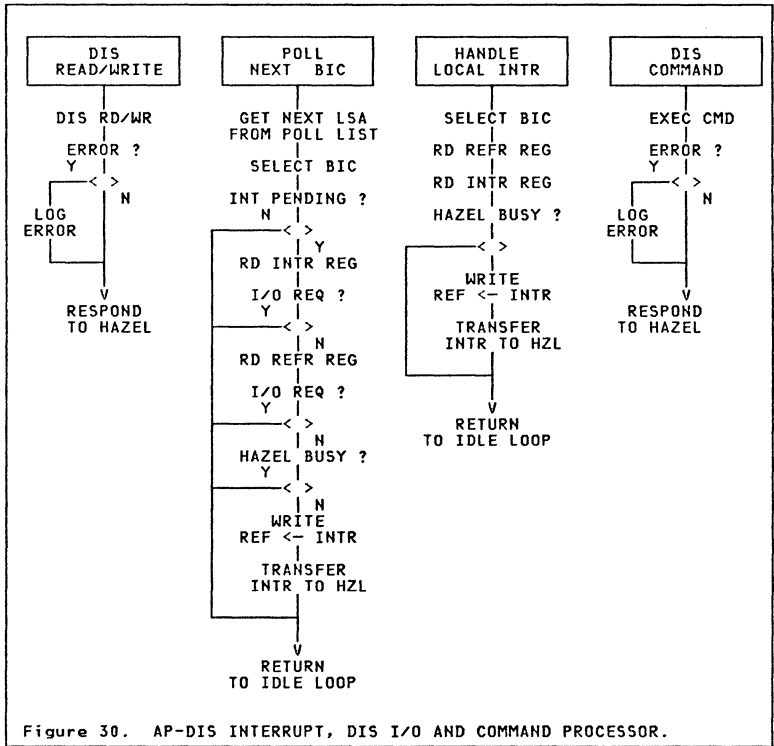
IBM Internal Use Only



As shown in Figure 29, the idle loop looks for new work. When new work is found, the idle loop passes control to the appropriate module, which after completing its function passes control back to the idle loop.

AP-DIS FUNCTIONS.

IBM Internal Use Only



RE-ISSUE HAZEL INTERRUPT.

Since Hazel does not latch AP interrupts the interrupts may be missed when:

- Hazel is running on a level lower than the level to which the interrupt was directed (level 3 in this case)
- Hazel has its interrupts masked off

It is the function of this code to detect such conditions and if they arise, re-issue a level 3 interrupt to Hazel.

DIS I/O REQUEST.

The I/O request may be a DISREAD, DISWRITE or DIS COMMAND.

Two different sets of modules are used to handle DISREAD/DISWRITE for local (parallel) and remote (serial) blocks. The read/write request is handled as follows:

IBM Internal Use Only

1. copy DCB from Hazel's memory to AP's registers
2. determine the 'flavour' of the request (e.g. bits I/O, immediate data)
3. execute read/write
4. if DIS error occurs log and abort execution even if the transfer count is not depleted
5. set return code in Hazel memory and interrupt Hazel on level 3 to signal completion of the I/O request

The DIS COMMANDs are:

- STOP POLLING
- START POLLING
- READ ERROR LOG
- READ POLL LIST
- UPDATE POLL LIST:
 - ADD LSA TO POLL LIST
 - REMOVE LSA FROM POLL LIST

An Error log reset request may be specified with all commands. The UPDATE POLL LIST command may require some interactions with the hardware. Every time a LSA is added to the poll list the following operations are performed to ensure that old interrupts are reset and that interrupts are enabled:

- select BIC
- read INTERRUPT register
- read REFERENCE register
- write REFERENCE <- INTERRUPT register (to reset pending interrupts)
- write CONTROL register <- enable interrupt

SERVICE LOCAL INTERRUPT.

When a macrofunction card in the local block generates an interrupt, level 3 is interrupted and sets a flag in a register. The idle loop checks this flag and jumps to LOCAL INTERRUPT code. The flowchart for this code is shown in Figure 30 on page 63.

POLL NEXT BIC.

Interrupts from macrofunction cards in the remote blocks do not interrupt AP. Therefore, AP must poll remote blocks to find the interrupts. AP does not poll all the remote blocks. It polls only the blocks that are in the poll list. The poll list is kept up to date by the operating system in Hazel. When a DSC program that requires macrofunction card interrupts is loaded or cancelled the operating system, if it is necessary, updates the poll list in AP using the DIS COMMAND.

MISCELLANEOUS ISSUES ASSOCIATED WITH INTERRUPT HANDLING.

An important design feature that is only applicable to the remote block, is the BREAK-AWAY from interrupt handling if a DIS I/O request was received by LEVEL 1. A check for DIS I/O request is made while the processor is waiting 50us between DIS operations associated with the interrupt handling (except during the 'WRITE INTERRUPT <- REFERENCE' operation). This feature reduces the impact of polling on the DIS I/O timings, by limiting the maximum delay before starting a DIS I/O command to a predictable value (about 120us).

A feature not shown in Figure 30 on page 63 is the handling of DIS errors while polling or handling interrupts. When an error occurs in any of the interrupt handling DIS operations the following is done:

IBM Internal Use Only

- error is logged
- control goes back to the idle loop

The INTERRUPT OVERRUN protection feature shown in the same figure requires further explanation. This mechanism stops AP from handling additional interrupts until Hazel has completed processing (POSTING) the previous interrupts. AP checks if Hazel is busy handling previous interrupts just before 'WRITE INTERRUPT <- REFERENCE' function is to be performed. If Hazel is still busy, the interrupt handling function is not carried any further and control is transferred to the idle loop.

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

RELIABILITY AND SERVICEABILITY

MAINTENANCE SUPPORT PLAN

- Test interface connectors on top of card for connection of diagnostic device.
- The card will be serviced as a replaceable FRU.
- The card will be repaired via Maintenance or vendor down to the component level.
- Mean Time to Replace - 15 minutes
- Mean Time to Repair - 2 hours
- Mean Time to Failure - 50,000 hours

ENVIRONMENT

Relative Humidity	8% to 80%	8% to 80%
Operating Temperature	10° to 43.3°C	50° to 110°F
Wet Bulb	26.7°C maximum	80°F maximum
Non-Operating Temperature	-65° to 150°C	
Forced Air Cooling		300 Linear Ft. per minute
Wet Bulb	26.7°C maximum	80°F maximum
Maximum Operating Altitude	7,000 feet above sea level	

POWER SUPPLIES

The APSCA requires the following voltages:

- +5 V 10% for all logic at approx. 3.0 amperes
- +8.5 V 5% for analog transceiver circuits at approx. .5 amperes
- 5 V 10% for analog receiver at approx 50 milliamperes.

The +1.7 V required for the granite modules is provided by an on-card regulator from the +5 V supply and the loading is approximately 2.2 amperes.

IBM Internal Use Only

TESTING

Nodal Impedance Test

The card requires a nodal impedance test (NIT) after assembly. This test will find any opens or shorts and placement problems and will guarantee that all components are wired and placed properly.

Functional Test

Once NIT is complete a functional test will be performed. This test will interact with the operator via Hazel A and the 3101 terminal. The testing will verify that communications to the RDIS from Hazel memory is established. Additional tests will verify other functions such as remote IPL, and the other functions implemented on the 9 pin 'd' shell.

Required test Hardware

1. ARIEL Basic Unit
2. Hazel A engine card
3. Hazel A memory card
4. APCSS Card
5. CSS-A Card
6. RG-62 AU 93 ohm test cable
7. Remote DIS unit

Required test Software

1. Firmware Application Code --resident on APSCA
2. EDXJ level interactive test software - ATST(BURNIN), \$DIS

Heading ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Heading References</u>
busdef	USINFO	8	ABUS Pin Definitions 5
cons	TECH	16	CONNECTORS 5
com	COMINT	23	APSCA/Remote DIS Communications 5
peer	COMINT	26	APSCA to APSCA (peer to peer) Communications 3
XR's	XRS	37	EXTERNAL REGISTER BIT ASSIGNMENTS

Figure ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Figure References</u>
sys	SYSARCH	2	1: 23
cdout	CDOUT	4	2:
rem	REMIPL	4	3: 3
dsh9	USINFO	5	4: 3
tpin	USINFO	6	5: 5
abus	USINFO	7	6:
test1	USINFO	11	7:
test2	USINFO	11	8:
test3	USINFO	12	9:
test4	USINFO	12	10:
normctc	NORMAL	14	11:
dataflo	DATAF	18	12: 17
addr	ADDRF	19	13: 17
LSRA	ORG	20	14: 20
XRasign	ORG	22	15: 21
iotime	COMINT	26	16: 25
cstime	COMINT	27	17: 25, 28
ppl	COMINT	29	18: 26
ppcom	COMINT	30	19: 27
clocks	CLK	33	20: 31
XRtime	XRS	36	21: 35
xrgr4	XRS	45	22: 57
xrint	XRS	46	23: 52, 53
lspr	XRS	47	24:
intlvl	INTSTRUC	49	25:
PSWfmt	INTSTRUC	51	26: 50
inthdwr	INTSTRUC	54	27: 52
FInst	MICROINS	60	28: 59
f1	APDIS	62	29: 62
f2	APDIS	63	30: 64, 64

Imbed Trace

Page iv	PREFACE
Page x	GENDESC
Page 2	SYSARCH
Page 2	USINFO
Page 3	CDOUT
Page 4	REMIPL
Page 12	USINFOF
Page 14	NORMAL
Page 14	TECH
Page 16	ORG
Page 17	DATAF
Page 18	ADDRF
Page 22	COMINT
Page 28	CLK
Page 32	XRS
Page 47	INTSTRUC
Page 53	ERRCHK
Page 56	DIAG
Page 57	MICROINS
Page 59	APDIS
Page 65	RAS

TPAC-A HARDWARE FUNCTIONAL SPECIFICATION

Control Systems Engineering

**IBM Corporation
General Technology Division
D-035 B-300-47A
East Fishkill
tel 533-7654**

IBM Internal Use Only

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

PREFACE

PREFACE

This document, describes the purpose of T-PAC, its capabilities, power requirements, how to order information, and a list of documents related to the card.

This guide also details the hardware needed, timing controls, external registers, pin assignments, wiring lists, and card programming.

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

General Description	1
Thorpe Programmable Application Card For Ariel (TPAC-A)	1
Abstract	1
Features	1
Features not on Thorpe Macrofunction Card	1
Features not on Thorpe Macrofunction Card or T-PAC	2
Ordering Information	2
Part Numbers and Related Documents	2
System Hardware Configuration	3
A-BUS Interface Definition	4
A-BUS Inputs and Outputs	4
Data Flow	5
Hardware Description	6
Read-Only Storage (ROS)	6
Microprogram Sequencer (PN 6832982)	7
Arithmetic Logic Unit (ALU) (PN 4745549)	8
Functional Storage Unit (FSU RAM) (PN 5120745)	8
Local Storage Register (LSR) (PN 8041469)	9
External Registers (PN 8494715)	9
Front End Transceiver (PN 2405058)	10
Clock Module (PN 2405060)	12
16 Bit Timer Module (PN 6017383)	14
Microcode	14
ROS Configuration Bits(0-17)	14
ROS Configuration Bits(18-48)	15
ROS Microinstruction Word Format	16
ROS Microinstruction Word, Bit Definitions	16
Microcode Instruction Cautions	19
Command Tag Combination (CTC)	19
External Register Assignments	20
External Register Addressing	20
External Register Bit Definitions	20
User Interface	24
D-Shells	24
Led's	25
Dip Switch	25
10 Position Tap Switch	26
Single Pole Double Throw Switches	27
Jumper Selection	28
Card Programming	28
Appendix A. Test Connectors	29

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

LIST OF ILLUSTRATIONS

Figure 1.	SECOND LEVEL DIAGRAM OF TPAC-A CARD	3
Figure 2.	TPAC-A ABUS 2nd LEVEL	4
Figure 3.	Data Flow among ALU, FSU, LSRs, and External Registers	6
Figure 4.	Microprogram Sequencer	8
Figure 5.	Block Diagram of External Register Module	9
Figure 6.	Receiving data via RS232 port (timing)	10
Figure 7.	Transmitting data via RS232 port (timing)	11
Figure 8.	Second Level Logic of Clock Module (on 2405060)	12
Figure 9.	Clock Cycle Representation. See	12
Figure 10.	Timing Diagram of Clock Module. Normal and memory access cycle	13
Figure 11.	R05 Bit Configuration Bits(18-48)	15
Figure 12.	R05 Word Configuration	16
Figure 13.	GROUP 0 -- External Register Module (U7)	21
Figure 14.	GROUP 1 -- External Register Module (U8)	22
Figure 15.	GROUP 2 -- External Register Module (U22)	23
Figure 16.	User Interface Pin Definition (25 Pin D-Shell)	24
Figure 17.	RS232 Communications Port Pin Definition, (9 Pin D-Shell)	24
Figure 18.	TEST CONNECTOR (JJ1)	29
Figure 19.	TEST CONNECTOR (JJ2)	29
Figure 20.	TEST CONNECTOR (JJ3)	30
Figure 21.	TEST CONNECTOR (JJ4)	30

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

GENERAL DESCRIPTION

THORPE PROGRAMMABLE APPLICATION CARD FOR ARIEL (TPAC-A)

Abstract

T-PAC (Thorpe Programmable Application Card) is a Customer Interface card for the Remote DIS (R-DIS) package. In essence it is a remap of the THORPE Macrofunction Card intended for use in high speed applications where an interface between the distributed interface system (RDIS, DIS) channel and external periphery is required. THORPE is a horizontal bit-slice microprocessor and is equipped with local storage registers, random access memory (RAM), external registers, a serial interface, a timer, and programmable read-only memory (ROM).

Features

- DIS Common I/O Interface
- 24 Digital Inputs
- 32 Digital Outputs
- Serial Communications Interface
 - EIA
 - Isolated Current Loop
- 2K x 8 FSU RAM
- 128 x 8 Local Storage Registers
- 1K x 48 Programmable ROM, Read-Only Storage (ROS)
- 250 ns Cycle Time, 666.66 ns Cycle Time for memory access
- 16 Bit Timer

Features not on Thorpe Macrofunction Card

- 16 bit down counter accessible via an external register
- Parity generated across DIS Bus (on card)
- Software programmable LED (signals LSA)
- Tx data is capable of being driven via ISO-L00P or EIA see Figure 17 on page 24
- New ALU module, all functions operational (same I/O configuration)
- Non-LSSD, testing via debug program and probing available test points
- RS232 port via 9-pin D-Shell instead of a 25-pin D-Shell
- Compass Drivers replace VTL 74S241
- Clear to Send and Request to Send are hardware programmable as common or independent signals
- The Digital Outputs are enabled via software control different than before.
 - Enable Xreg6, Xreg7 will enable registers 6 and 7 only, and
 - Enable XregA, XregB will enable registers A and B. These two control lines replace the - Enable and + Enable lines of THORPE respectively.

Note: This is of prime importance when swapping microcode between the THORPE and T-PAC

Note: This card should be used along-side another extender card for full use. This Extender Card will allow all 24 DI's and 32 DO's of T-PAC to the user. Without this card only 16 DI's and 8 DO's will be available to the user, but ** it is not necessary **.

Note: In this User Guide it should be noted that for all purposes a logical 0 is ground and a logical 1 is +5v.

IBM Internal Use Only

Features not on Thorpe Macrofunction Card or T-PAC

- 2K x 48 RDS Address capability
- 4 LED's available to software for diagnostics

Ordering Information

The responsible engineering department is Process Control Systems Engineering, Department 035, East Fishkill, N.Y.

Part Numbers and Related Documents

Part Numbers are:

- Assembly -> 6912778
- Raw Card -> 6912779
- Schematic -> 6912780
- EC -> A37493K

Related documents are:

- Auto Poller Functional Specification, E45-1339, PN 4745305
- DSS User Guide Volume I, Z45-1123-0, PN 6856716
- DSS User Guide Volume II, Z45-1124-0, PN 4745244
- THORPE Design Information Technical Report, TR-74.023
- THORPE Microcode Development User's Guide, E45-1242, PN 4745312

IBM Internal Use Only

SYSTEM HARDWARE CONFIGURATION

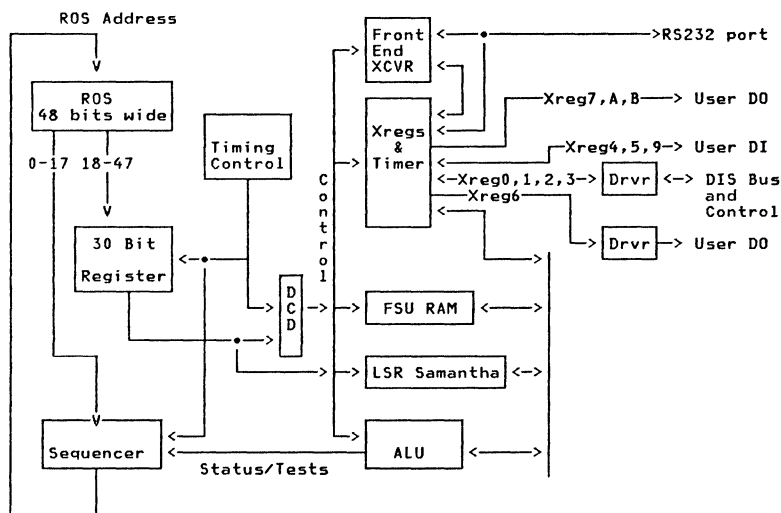


Figure 1. SECOND LEVEL DIAGRAM OF TPAC-A CARD

The ROS (Read Only Storage) is the microcode that describes the function of Tpac-A. Presently there are three different functions for Tpac-A. They are Auto-Poller, Pulsed Output Modulation (POM), and Semiconductor Equipment Communications Standard (SECS). The first 18 bits of ROS are used along with status signals from the ALU to generate the next instruction address. This is accomplished by the Sequencer which has inputs of 18 ROS signals, 13 A Test, and 13 B Test signals which come from the ALU. The determination of which signals to use for the next address depends on the outcome condition of the previous instruction.

The 30 bit register latches the remaining 30 bits (18-47) of the ROS word. These signals will be used for control throughout all parts of Tpac-A; such as ALU control, XR address, and Emit Bits.

The timing control module generates the clocks for the system, and the FSU (2Kx9) RAM select and reset. Along with this function, it generates the LSR address from either the Emit bits or the A REG of the ALU, and also latches ALU data for the LSR data bus. The Transceiver module generates to and receives from analog circuitry the RS232 data. It also generates control signals to LSR's (Samantha), XR (External Registers), FSU, and the ALU. The remaining components of Tpac-A are the External Registers, Samantha, and FSU RAM. The External Registers allow an interface to the DIS Bus, and I/O. Samantha is used for Local Storage Registers, and FSU is implemented as a 2Kx9 data buffer.

All of these components will be described in more detail further on in the spec.

IBM Internal Use Only

A-BUS INTERFACE DEFINITION

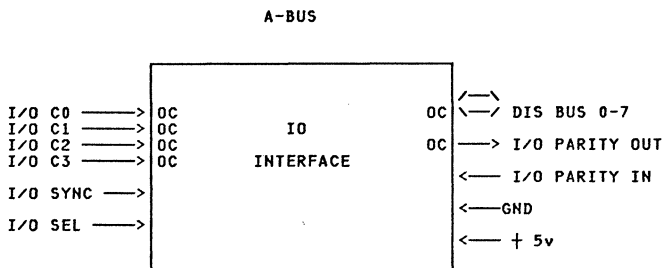


Figure 2. TPAC-A ABUS 2nd LEVEL: The DIS channel is an open-collector bus. The terminators for the OC signals will reside on the board.

A-BUS Inputs and Outputs

The A-Bus will provide the path for data transfer to/from the Tpac-A card. All control, Data Busses, Address, and Power will be provided from the A-Bus, except the Program Voltage (VPP).

The following is a list of all the A-Bus I/O on Tpac-A and their definitions as a function of the Hazel-A card. These I/O are shown in Figure 2.

- IO SYNC OUT I DIS Sync ; When active, signals data and CTC's are valid on the DIS Bus to the Tpac-A card provided IO Select is active.
- IO RETURN IN O DIS Return ; When reading Tpac-A, this output signals data is valid on the bus. When writing Tpac-A, this output acknowledges the data has been received and is valid, (ie. good parity). It is enabled when Sync, and IO Sel are active during a DIS Read, and Sync, IO Sel, and parity is valid on the DIS Bus during a DIS Write.
- IO SELECT I DIS LSA Select ; This input Enables Tpac-A. When active, it will allow the DIS Bus, C0-C3, Sync, and Return through it's buffers.
- C0,C1,C2,C3 I Command Tag Combination ; These inputs are used as control signals to Tpac-A. They are valid when IO Sel, and Sync are valid.
- IO D0-D7 I/O DIS Bus Bits 0-7 ; These bidirectional open-collector signals are the data bus to/from the Tpac-A card. When reading Tpac-A, data is valid before Return becomes active. When writing Tpac-A, data is valid before Sync becomes active.
- IO DP IN O DIS Data Parity IN ; This output is generated to guarantee Negative Active ODD Parity across the DIS Bus (ie. an odd number of gnds on the DIS Bus and IO DP IN)

IBM Internal Use Only

IO DP OUT **I** DIS Data Parity Out ; This input and the DIS Bus are checked to guarantee Negative Active ODD Parity during a DIS Write. If Parity is bad (ie. not an odd number of gnds across the 9 bits), then return will not be generated.

Voltages **I** There are 9 different voltage levels available on the Abus connector. They are;

Voltages		Pins	
1.	gnd	116, 118, 120	
2.	+ 5v	115, 117, 119	3.2 ma max
3.	+ 8.5v	33	200 ma max
4.	+ 12v	48	50 ma max
5.	- 12v	94	25 ma max
6.			
7.	- 5v	71	not used
8.	+ 15v	1	not used
9.	- 15v	89	not used
10.	Agnd	2	not used

Note: Agnd is Analog ground and should be kept separate from gnd (digital ground).

DATA FLOW

The data flow among the ALU, FSU, LSRs and the External Registers are detailed in Figure 3 on page 6

IBM Internal Use Only

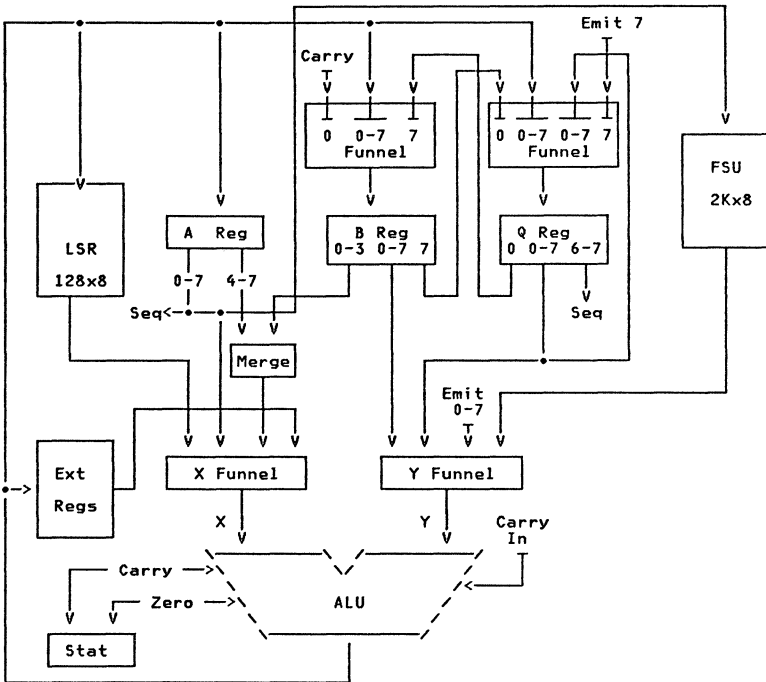


Figure 3. Data Flow among ALU, FSU, LSRs, and External Registers

The X Funnel is the data path to the ALU of the External registers, LSR data, A reg (ALU), or bits 0-3 of the B REG (ALU). The Y Funnel allows the B REG (ALU), Q REG (ALU), Emit data field, and the FSU data to the ALU. the B REG and the Q REG allows data to be shifted right or left, and the ALU output is available to the External Registers, LSRs, A REG, B REG (thru Funnel), Q REG (thru Funnel), and FSU (thru the A REG).

HARDWARE DESCRIPTION

Refer to Figure 1 on page 3

Read-Only Storage (ROS)

Tpac-A can use a 1K x 8 or a 2K x 8 PROM to implement the ROS

1. 1K x 8 --> P/N 4429933
2. 2K x 8 --> P/N 5616833

IBM Internal Use Only

At present there are three applications for Tpac-A they are;

1. Auto-Poller
2. SECS and
3. Pulsed Output Modulation

These are not documented in this spec. Each has their own spec.

Each R0S word controls the data flow of the control logic in one normal clock cycle of 250 ns if FSU is not involved, or 666.666 ns if FSU is involved. See Figure 9 on page 12. The first 18 bits of R0S are used by the Sequencer along with test bits from the ALU to determine the next address of R0S to be executed. The remaining 30 bits are latched through the 30 bit register and are control signals through-out the whole system.

Microprogram Sequencer (PN 6832982)

Takes the first 18 bits of a R0S word, contents of A and Q registers, and the status of the arithmetic logic unit (ALU) to determine the address of the next R0S word. See Figure 4 on page 8.

IBM Internal Use Only

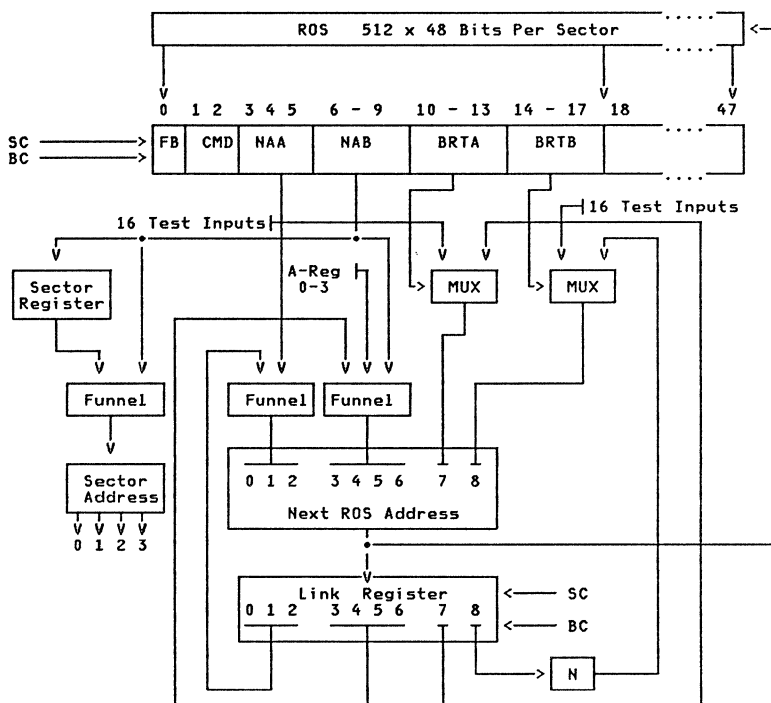


Figure 4. Microprogram Sequencer: Determines the Next Address for the R0S. Uses first 18 bits of the R0S microinstruction word.

Arithmetic Logic Unit (ALU) (PN 4745549)

An 8-bit ALU that performs arithmetic and logic functions under the control of the ALU field of the R0S word.

Functional Storage Unit (FSU RAM) (PN 5120745)

A 2K x 8-bit buffer accessible by the ALU.

IBM Internal Use Only

Local Storage Register (LSR) (PN 8041469)

128 x 8 Local Storage Registers accessible by the ALU. They are contained in one Samantha module and are general purposes registers that can be used for counters, pointers, status, and control registers.

External Registers (PN 8494715)

Used for I/O data, Communication interface, and Control. There are six read-writable 8-bit registers and six readable registers contained in three GOLF modules.

See Figure 5 for second level logic of the external register and "External Register Addressing" on page 20 for the register addressing scheme.

Xreg 4,5,9	24 Digital Inputs	User Interface
Xreg 6,7,A,B	32 Digital Outputs	User Interface
Xreg 0,1	16 Digital Inputs	DIS Bus Interface
Xreg 2,3	16 Digital Outputs	DIS Bus Interface
Xreg 8	8 Digital Inputs	R5232 Communications Port Interface
Total	96 Digital I/O	

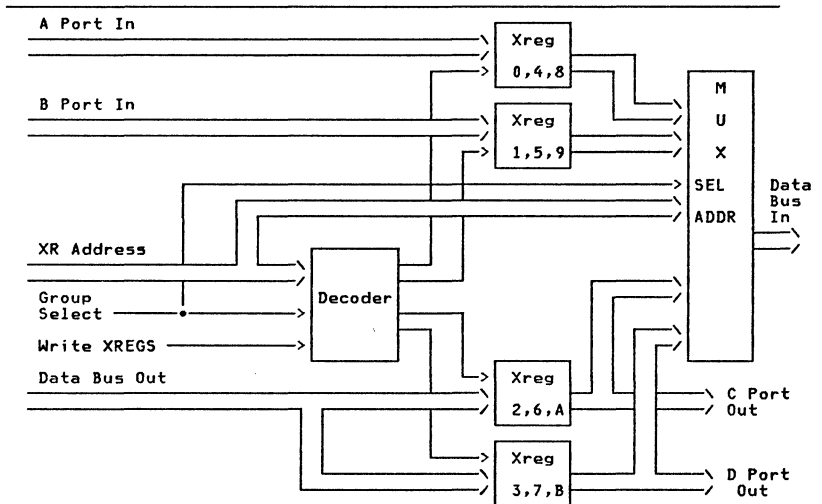


Figure 5. Block Diagram of External Register Module: The numbers represent the corresponding register defined in "External Register Assignments" on page 20.

IBM Internal Use Only

Front End Transceiver (PN 2405058)

The Transceiver module provides a programmable interface for serial TP communication. The module supports the START-STOP protocol and provides all the necessary lines for connection to an RS232 interface.

The Front End Transceiver interfaces with the analog circuit. It generates a clock at a rate twice that of the baud rate of transmission. The clock generator is inactive in non-transmission mode.

When serial data comes in, the Start bit (Space) activates the clock generator. When the control program detects the clock is ON, it has to turn on an External Register bit before the Start bit ends, to keep the clock generator active. When the clock generator is active, the clock output will be on from the mid-points of the bits to the start of the subsequent bits.

It is the responsibility of the control program to keep count of the incoming bits and to turn Off the External Register bit that activates the clock generator after the 'STOP' bit.

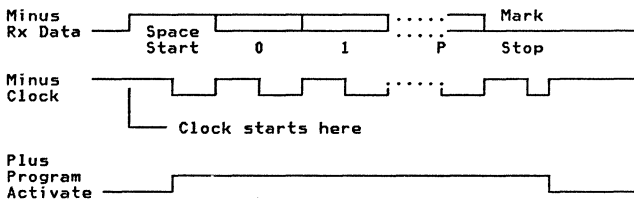


Figure 6. Receiving data via RS232 port (timing): RTS and CTS maybe hardware jumpered to be common or independent signals. See the Schematic

In this way, the clock is synchronized with the data being received and the control program knows the data bit is stabilized whenever the clock is On. See Figure 6

When the control program wants to transmit data, it puts the transmit data bit in bit 0 of the addressed LSR with ROS bit 34 On in one microinstruction. The transmit data will be temporarily stored in the transceiver. Then the control program will turn on an External Register bit reserved for Request-To-Send (RTS) which caused Clear-To-Send (CTS) to be On from the data set. (For in-house arrangement, RTS will be tied with CTS). CTS activates the clock generator. When the clock is On, it will bring out the Transmit Data bit at the output of the transceiver.

When the control program detects that the clock went down, it dumps the next data bit from LSR bit 0 into the transceiver and repeats the same cycle until the whole data byte and Stop bit are transmitted. Then it turns Off the RTS which eventually inactivates the clock generator.

IBM Internal Use Only

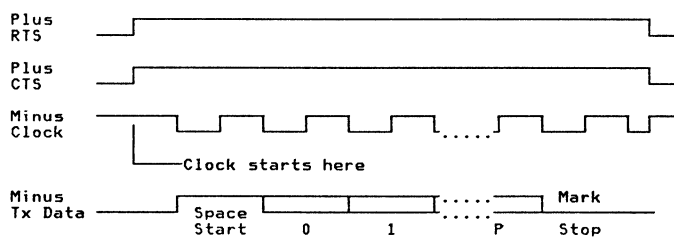


Figure 7. Transmitting data via RS232 port (timing)

IBM Internal Use Only

clock Module (PN 2405060)

Using a 24 MHz oscillator This Golf Module generates all system clocks, for normal operation and memory access (see Figure 10 on page 13, and Figure 9) it supplies the logic for the LSR interface, including address lines and data lines, it selects and resets the FSU and also generates a 1 MHz square wave for the timer module (pn 6017383), see Figure 8

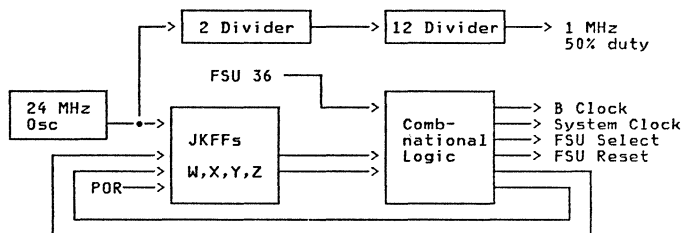


Figure 8. Second Level Logic of Clock Module (pn 2405060)

	W	X	Y	Z	
0	0	0	0	0	
1	0	0	0	1	
2	0	0	1	1	
3	0	0	1	0	
4	0	1	1	0	
5	1	1	1	0	
6	1	0	1	0	
7	1	0	1	1	
8	1	0	0	1	
9	1	0	0	0	
10	1	1	0	0	
11	1	1	0	1	
12	1	1	1	1	
13	0	1	1	1	
14	0	1	0	1	
15	0	1	0	0	

$$\text{FSU Select} = Y\bar{Z} + W\bar{X} + W\bar{Y}$$

$$\text{System Clock} = \bar{W} \times Z$$

$$\text{FSU RESET} = Y\bar{Z}$$

$$B \text{ Shift Clock} = \overline{W} \overline{X} \overline{Y}$$

Figure 9. Clock Cycle Representation. See Figure 10 on page 13

IBM Internal Use Only

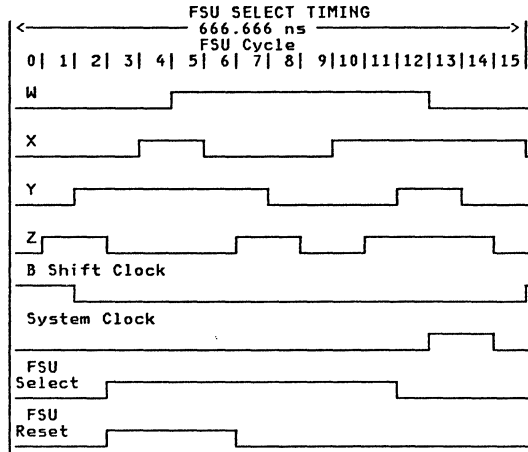
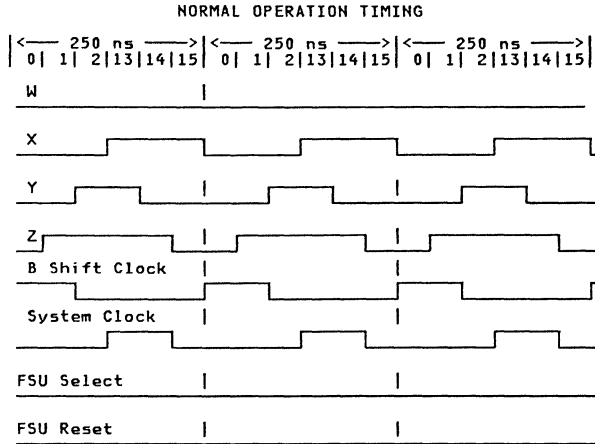


Figure 10. Timing Diagram of Clock Module. Normal and memory access cycle
See Figure 9.

IBM Internal Use Only

16 Bit Timer Module (PN 6017383)

A hardware programmable counter accessible via Xreg9. "User Interface" on page 24 describes how the counter output can be accessed via Xreg9.

Also generates parity for the R-DIS Data Bus.

MICROCODE

Each microinstruction word is 48 bits long. The first 18 bits determine the next microinstruction address, Figure 4 on page 8 describes the hardware assignment of the first 18 bits of the microinstruction word. The last 30 bits control the data flow among the ALU, LSRs, FSU, and External Registers and Figure 11 on page 15 describes the hardware assignment of the these microinstruction bits

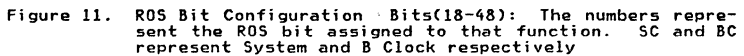
Figure 12 on page 16 describes the format of the microinstruction word.

ROS Configuration Bits(0-17)

See Figure 4 on page 8 for the hardware bit assignments of the first 18 bits of the microinstruction word.

These 18 bits determine the next microinstruction address as described in "ROS Microinstruction Word, Bit Definitions" on page 16.

ROS Configuration Bits(18-48)



IBM Internal Use Only

R0S Microinstruction Word Format

0 0	0 1	0 2	0 3	0 4	0 5	0 6	0 7	0 8	0 9	1 0	1 1	1 2	1 3	1 4	1 5	1 6	1 7	
FB	CMD	NAA		NAB						BRTA				BRTB				

1 8	1 9	2 0	2 1	2 2	2 3	2 4	2 5	2 6	2 7	2 8	2 9	3 0	3 1	3 2	3 3	3 4	3 5	3 6	3 7	3 8	3 9	4 0	4 1	4 2	4 3	4 4	4 5	4 6	4 7
		XFC	YFC	SFC		ALU				C I	Z A	Z B	Z Q		Misc	FSU	XR	S											Emit

Figure 12. R0S Word Configuration: For hardware assignments of the microinstruction word, bits see Figure 4 on page 8 and Figure 11 on page 15.

R0S Microinstruction Word, Bit Definitions

The following will describe in detail the function of all bits in the microinstruction word.

R0S BITS (0-17) FOR GENERATION OF NEXT ADDRESS		
NAME AND (R0S BITS)	DECODED AS	DESCRIPTION
FUNCTION BRNACH, FB (0)	0	Allows NAB field of microinstruction to generate bits 3-6 of next R0S address
	1	Allows bits 0-3 of A Reg to generate bits 3-6 of next R0S address. NAB field is loaded in the Sector Register. This value becomes the address for subsequent microinstructions.
COMMAND CMD (1,2)	00 NOP	Executes next microinstruction as per next R0S address
	01 RPT	Executes current microinstruction twice. First time through, it inhibits destination registers A, B, or Q from accepting new data.
	10 LNK	Executes next microinstruction as per next R0S address. Saves the current R0S address in the Link Register.
	11 RTN	If the Link Register is holding an address, as a result of a previous link instruction, it then generates the next R0S address from the value in the Link Register after inverting the least significant bit (LSB). It then releases the link register
NEXT ADDRESS A, NAA		These three bits generate bits 0-2 of the next R0S address

IBM Internal Use Only

(3-5)		
NEXT ADDRESS B, NAB (6-9)		These four bits generate bits 3-6 of the next ROS address under normal microinstruction execution. When the Function Branch bit is on, the value in the NAB field is loaded into the Sector Register and becomes the new Sector address
BRANCH TEST A, BRTA (10-13)	0000- 0 0001- 1 0010- C 0011- Z 0100- 0101- A0 0110- A1 0111- A2 1000- A3 1001- Q6 1010- X0 1011- X1 1100- X2 1101- X3 1110- 1111- 0	These four bits are used to determine bit 7 of the next ROS address Forced to zero Forced to one One if carry stat is on One if Zero stat is on Spare One if A reg bit 0 is on One if A reg bit 1 is on One if A reg bit 2 is on One if A reg bit 3 is on One if Q reg bit 6 is on Selected Ext Reg bit 0 Selected Ext Reg bit 1 Selected Ext Reg bit 2 Selected Ext Reg bit 3 Spare Forced to zero
BRANCH TEST B, BRTB (14-17)	0000- 0 0001- 1 0010- C 0011- Z 0100- 0101- A4 0110- A5 0111- A6 1000- A7 1001- Q7 1010- X4 1011- X5 1100- X6 1101- X7 1110- PE 1111- 0	These four bits are used to determine bit 8 of the next ROS address Forced to zero Forced to one One if carry stat is on One if Zero stat is on Spare One if A reg bit 4 is on One if A reg bit 5 is on One if A reg bit 6 is on One if A reg bit 7 is on One if Q reg bit 7 is on Selected Ext Reg bit 4 Selected Ext Reg bit 5 Selected Ext Reg bit 6 Selected Ext Reg bit 7 Parity Error Forced to zero

IBM Internal Use Only

ROS BITS (18-48) FOR DATA FLOW CONTROL			
NAME AND (ROS BITS)	DECODED AS	DESCRIPTION	
X FUNNEL CONTROL, XFC (18,19)	00- XR 01- LSR 10- SL4 11- A	Selects one of four possible sources for the X side of the ALU External Register to X side Samantha Ram Buffer to X side Bits 4-7 of A register become the high order nibble and bits 0-3 of the B register become the lower order nibble of the ALU X side A Register to X side	
Y FUNNEL CONTROL, YFC (20,21)	00- Q 01- E 10- FSU 11- B	Selects one of four possible sources for the Y side of the ALU Q register to Y side Emit field of microinstruction to Y side FSU Ram Buffer to Y side B register to Y side	
SHIFT FUNNEL CONTROL, SFC (22,23)	00- 01- 10- SL1 11- SR1	Selects one of three possible sources for the B and Q registers Selects output of the ALU for both the B and Q Registers Same as above ALU output bits 1-7 go to B register bits 0-6, Q register bit 0 goes to B register bit 7, and ALU bit 0 is lost Q register bits 0-6 get previous Q register bits 1-7, and Q register gets Emit field bit 7 ALU bits 0-6 go to B register bits 1-7, Carry from ALU goes to B register bit 0, ALU bit 7 goes to Q register bit 0, and Q register bits 1-7 get the previous Q register bits 0-6. Q register bit 7 is lost	
		CARRY IN = 0	CARRY IN = 1
ARITHMETIC LOGIC UNIT ALU (24-27)	0000- AND 0001- COI 0010- ONE 0011-PASSY 0100-ZERO 0101-PASSX 0110- EOR 0111- OR 1000- X+X 1001- X+Y 1010- X+1 1011- X-1 1100- Y+1 1101- Y-1 1110- X-Y 1111- Y-X	X and Y X coincidence Y Force all ones Flush Y side Force all zeros Flush X side X exclusive-or Y X or Y X plus X X plus Y not used not used not used not used X minus Y minus 1 Y minus X minus 1	X and Y X coincidence Y Force all ones Flush Y side Force all zeros Flush X side X exclusive-or Y X or Y X plus X plus 1 X plus Y plus 1 X plus 1 X minus 1 Y plus 1 Y minus 1 X minus Y Y minus X
CARRY IN + CI (28)	0 1	Used with the ALU operations shown above No carry Carry is forced	
+ ZA (29)	0 1	Inhibits A register from accepting data Allows A register to accept data	
+ ZB (30)	0 1	Inhibits B register from accepting data Allows B register to accept data	
+ ZQ (31)	0 1	Inhibits Q register from accepting data Allows Q register to accept data	
NOTE : ZA, ZB, ZC may be used in any combination to allow multiple settings of registers during a single cycle			

IBM Internal Use Only

MISC 0 (32)	0 1	Writes to Samantha RAM
MISC 1 (33)	0 1	Writes to External registers
MISC 2 (34)	0 1	Writes LSR bit 0 to Transmitter
FSU RD/WR and XR Group Sel (35,36)	00-XR(##) 01-FS(R) 10-XR(##) 11-FS(W)	Select XR address # = 0,1,2,3 if Emit bit 0=0, or # = 8,9,A,B if Emit bit 0=1 Read from FSU RAM Select XR address # = 4,5,6,7 if Emit bit 0=0, of # = C,D,E,F if Emit bit 0=1 Writes To FSU RAM
XR ADDRESS (37,38)	00 01 10 11	External Register # = 3,7,B,F External Register # = 2,6,A,E External Register # = 1,5,9,D External Register # = 0,4,8,C
SELECT LSR ADDR Source (39)	0-LS(E) 1-LS(A)	Emit field is used to address the LSR A register is used to address the LSR
NOTE the use of the Emit field depends on the other fields in the microinstruction		
(40-47) (47)	0 - 7 7	Used as immediate data on the ALU Y side Used as the serial input to the Q register's lowest order bit in a shift left operation
(41-47)	1 - 7	Used as the address for the LSR's; Bit 1 - selects one of two groups in the Samantha module Bit 2-7 - selects one of 64 LSR's in the group
(40)	0 1	Used with R05 bit 35 of the FSU field to select the desired External register Select XR address # = 0,1,2,3 if R05 bit 35=0, or # = 4,5,6,7 if R05 bit 35=1 Select XR address # = 8,9,A,B if R05 bit 35=0, or # = C,D,E,F if R05 bit 35=1

Microcode Instruction Cautions

Notes:

1. Emit Bit 0 will automatically be set by the assembler according to which XR has been selected.
2. Avoid using the Emit field for immediate data while executing an XR instruction, as Emit bit 0 will be used to select the XR.
3. However, if necessary, immediate data can be specified with the following restrictions. When selecting XRs 0 thru 7, only a positive number 00-7F can be used in the Emit field; when selecting XRs 8 thru F, only a negative number 80-FF can be used in the Emit field.
4. Failure to do this will result in the wrong XR being selected.

COMAND TAG COMBINATION (CTC)

The CTC's are non standard and can be defined via the Microcode written by the user. There are, however, two restrictions listed below;

IBM Internal Use Only

1. CTC 0 -> Microcode must be recognized as an LSA Select
2. CTC F -> Microcode must be recognized as a Read Macrofunction ID

EXTERNAL REGISTER ASSIGNMENTS

There are a total of three External Register modules. Each module is capable of 16 Digital Inputs and 16 Digital Outputs, and are addressed as follows;

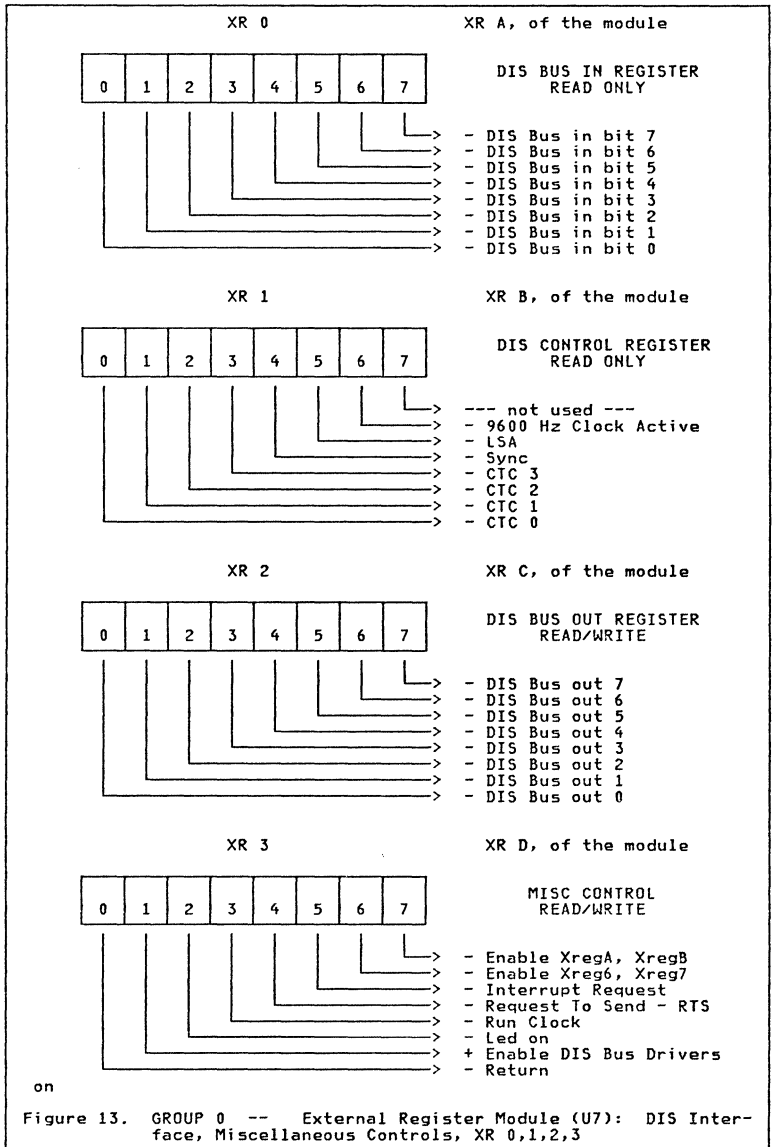
External Register Addressing

ROS Bit 40 Emit Bit 0	ROS Bit 35 +Rd/-Wr/ Xr Sel	0	0	1	1	ROS Bit 37 XR Adr bit 0
		0	1	0	1	ROS Bit 38 XR Adr bit 1
1	1	Xreg3	Xreg2	Xreg1	Xreg0	
1	0	Xreg7	Xreg6	Xreg5	Xreg4	
0	1	XregB	XregA	Xreg9	Xreg8	
0	0	not used				

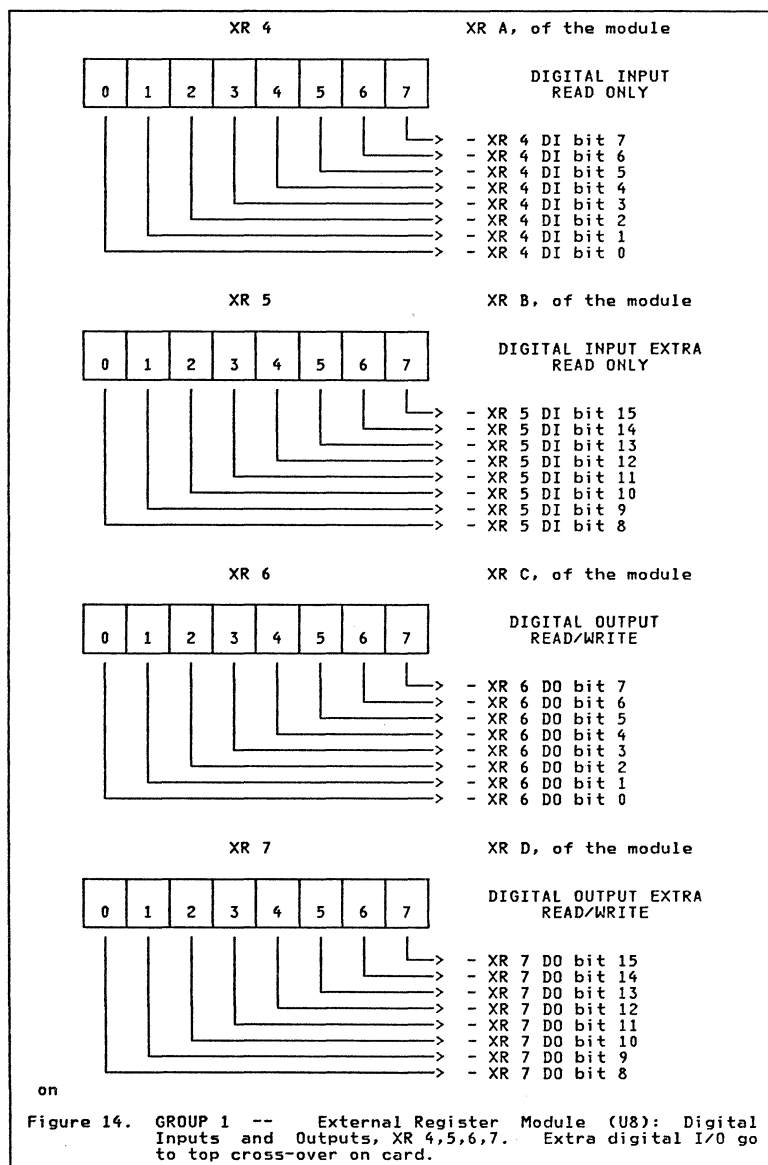
External Register Bit Definitions

Note: - Enable Xreg6, Xreg7 and - Enable XregA, XregB differ from - Enable and + Enable of the THORPE card. - Enable Xreg6, Xreg7 enables Xreg6 and 7 while - Enable XregA, XregB enables XregA and B. - Enable and + Enable of THORPE were both used to enable xreg6 and xreg7

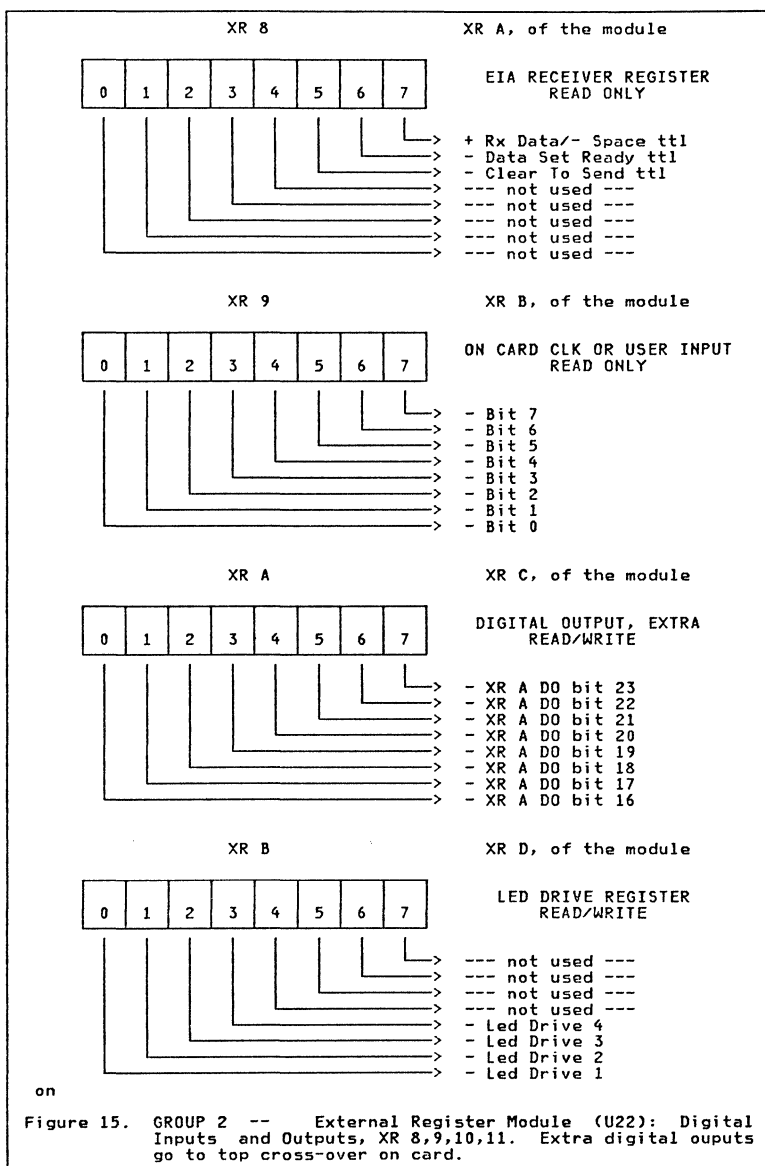
IBM Internal Use Only



IBM Internal Use Only



IBM Internal Use Only



IBM Internal Use Only

USER INTERFACE

D-Shells

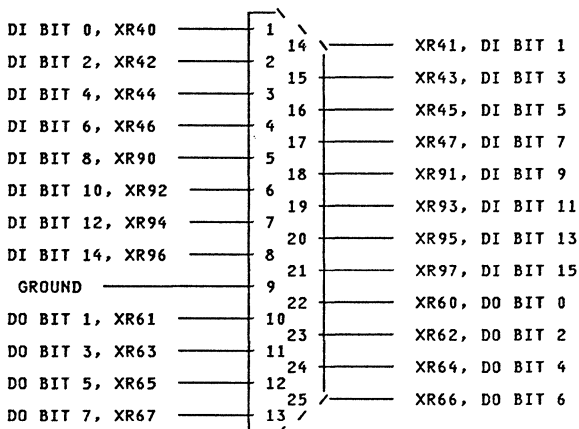


Figure 16. User Interface Pin Definition (25 Pin D-Shell)

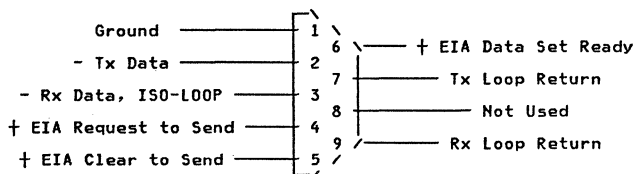


Figure 17. RS232 Communications Port Pin Definition, (9 Pin D-Shell)

Note: The Tx data is hardware programmable as either EIA Tx Data or Iso-Loop Tx data via a jumper pin. See the Schematic.

Led's

There are 5 leds on card. One is available to the user through the front plate on the card. This led is available through XR 3 bit 3. The other 4 leds are packaged as a group and are placed near the top of the card. They are not available through the front plate of the card but can be used for diagnostics. They are available through XR B bits 0, 1, 2, and 3.

IBM Internal Use Only

Dip Switch

This switch controls miscellaneous hardware functions on the card. It should be noted that once set up they need not be altered unless for debug purposes.

Switch 1 Selects high/low byte of Timer module counter to Xreg9. See "10 Position Tap Switch" on page 26 for selection of actual counter bit.

1. ON - low - Enables high byte of counter to mux
2. OFF - high - Enables low byte of counter to mux

Switch 2, 3 Enables modem operation of RS232 port

1. 2 ON and 3 OFF - Request to Send (RTS) tied to Clear to Send (CTS)
2. 2 OFF and 3 ON - Request to Send separate from Clear to Send

Switch 4, 5 Not Used

Switch 6 Controls EIA receiver threshold of + EIA Data Set Ready

1. ON - high - EIA receiver operates in normal mode
2. OFF - open - EIA receiver operates in fail-safe mode

Switch 7 Controls EIA receiver threshold of + EIA Clear to Send

1. ON - high - EIA receiver operates in normal mode
2. OFF - open - EIA receiver operates in fail-safe mode

Switch 8 Controls EIA receiver threshold of + Rx Loop Test

1. ON - high - EIA receiver operates in normal mode
2. OFF - open - EIA receiver operates in fail-safe mode

Note: Pin 14 of the EIA receiver is left open to operate in the fail-safe mode since that signal is already controlled by Switch 8

Note: For definition of Normal and Fail-Safe modes see 'The Line Driver and Line Receiver Data Book' from Texas Instruments under vendor PN 75154

IBM Internal Use Only

10 Position Tap Switch

This Switch controls which clock output of the Timer module will be available to the multiplexer of Xreg9.

Note: See Switch 1 in "Dip Switch" on page 25 for selection of high/low byte of counter.

Position 1	Bit 0/8 to Xreg9 Hz	393208/1536	usec	2.543/651.04167
Position 2	Bit 1/9 to Xreg9 Hz	196608/768	usec	5.086/1302.0833
Position 3	Bit 2/10 to Xreg9 Hz	98304/384	usec	10.1725/2604.1667
Position 4	Bit 3/11 to Xreg9 Hz	49152/192	usec	20.3450/5208.3333
Position 5	Bit 4/12 to Xreg9 Hz	24576/96	usec	40.6901/10416.667
Position 6	Bit 5/13 to Xreg9 Hz	12288/48	usec	81.3802/20833.333
Position 7	Bit 6/14 to Xreg9 Hz	6144/24	usec	162.760/41666.667
Position 8	Bit 7/15 to Xreg9 Hz	3072/12	usec	325.5208/83333.333
Position 9	+ 5 volts to Xreg9			
Position 10	Gnd to Xreg9			

Note: Count 0-7 is the high byte, thus lower frequency

Note: All counter outputs from the Timer module are 50 % duty cycle

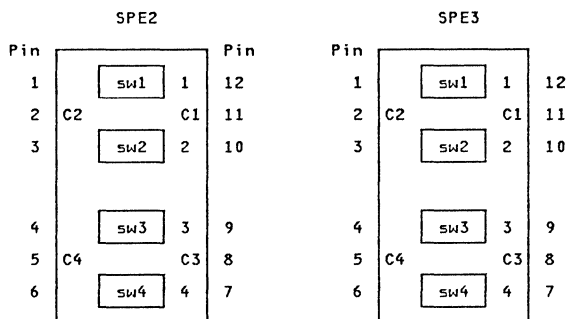
IBM Internal Use Only

Single Pole Double Throw Switches

These switches multiplex the on card counter frequency and DI bits 8 - 15 to Xreg9.

Switch to the left —> On card counter frequency to Xreg9

Switch to the right —> Digital Input to Xreg9



Switch	switch to right	switch to left	
U28			
sw1	counter freq	DI bit 8	-> to Xreg9 bit 0
sw3	counter freq	DI bit 9	-> to Xreg9 bit 1
sw4	counter freq	DI bit 10	-> to Xreg9 bit 2
sw2	counter freq	DI bit 11	-> to Xreg9 bit 3
U29			
sw1	counter freq	DI bit 12	-> to Xreg9 bit 4
sw3	counter freq	DI bit 13	-> to Xreg9 bit 5
sw4	counter freq	DI bit 14	-> to Xreg9 bit 6
sw2	counter freq	DI bit 15	-> to Xreg9 bit 7

The side of the switch which is down defines left or right in the table above

IBM Internal Use Only

Jumper Selection

This jumper selects the type of transmit data for the RS232 port. The Transmit data can be driven by an Isolated Current Loop or EIA.

- ZM1
- POS 1 [
- ZM2
- POS 2 [
- ZM3

Position 1 → Selects Transmit Data for Isolated Current Loop
Position 2 → Selects Transmit Data for EIA Standard

Note: These pins are located directly behind the 9 pin D-Shell

CARD PROGRAMMING

For detailed information relating to card programming, refer to the THORPE Microcode Development User's Guide, E45-1242, PN 4745312.

IBM Internal Use Only

APPENDIX A. TEST CONNECTORS

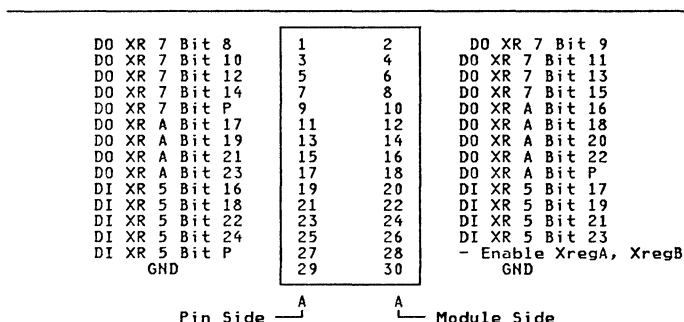


Figure 18. TEST CONNECTOR (JJ1): Used as a crossover to extender card for more I/O

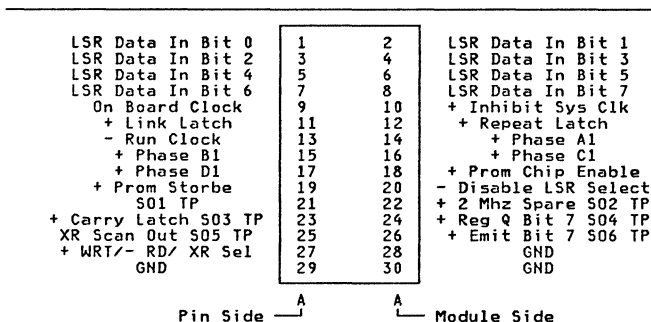


Figure 19. TEST CONNECTOR (JJ2): S0 => Scan Out

IBM Internal Use Only

+ ALU Out Bit 0	1	2	+ ALU Out Bit 1
+ ALU Out Bit 2	3	4	+ ALU Out Bit 3
+ ALU Out Bit 4	5	6	+ ALU Out Bit 5
+ ALU Out Bit 6	7	8	+ ALU Out Bit 7
+ REG A Bit 0	9	10	+ REG A Bit 1
+ REG A Bit 2	11	12	+ REG A Bit 3
+ REG A Bit 4	13	14	+ REG A Bit 5
+ REG A Bit 6	15	16	+ REG A Bit 7
+ REG B Bit 0	17	18	+ REG B Bit 1
+ REG B Bit 2	19	20	+ REG B Bit 3
+ REG B Bit 4	21	22	+ REG B Bit 5
+ REG B Bit 6	23	24	+ REG B Bit 7
- Select LSR Odd	25	26	- Select LSR Even
- Write To LSR	27	28	- Select FSU Even
- Select FSU Odd	29	30	- FSU Reset

Pin Side A A Module Side

Figure 20. TEST CONNECTOR (JJ3)

+ REG Q Bit 0	1	2	+ REG Q Bit 1
+ REG Q Bit 2	3	4	+ REG Q Bit 3
+ REG Q Bit 4	5	6	+ REG Q Bit 5
+ REG Q Bit 6	7	8	+ REG Q Bit 7
+ LSR Data Out Bit 0	9	10	+ LSR Data Out Bit 1
+ LSR Data Out Bit 2	11	12	+ LSR Data Out Bit 3
+ LSR Data Out Bit 4	13	14	+ LSR Data Out Bit 5
+ LSR Data Out Bit 6	15	16	+ LSR Data Out Bit 7
+ FSU Data Bit 0	17	18	+ FSU Data Bit 1
+ FSU Data Bit 2	19	20	+ FSU Data Bit 3
+ FSU Data Bit 4	21	22	+ FSU Data Bit 5
+ FSU Data Bit 6	23	24	+ FSU Data Bit 7
+ LSR Adr 0 - 32	25	26	+ LSR Adr 1 - 16
+ LSR Adr 2 - 8	27	28	+ LSR Adr 3 - 4
+ LSR Adr 4 - 2	29	30	+ LSR Adr 5 - 1

Pin Side A A Module Side

Figure 21. TEST CONNECTOR (JJ4)

Heading ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Heading References</u>
gendoc	TPACA1	1	General Description
tpac	TPACA1	1	Thorpe Programmable Application Card For Ariel (TPAC-A)
abst	TPACA1	1	Abstract
feature	TPACA1	1	Features
addfeat	TPACA1	1	Features not on Thorpe Macrofunction Card
feat	TPACA1	2	Features not on Thorpe Macrofunction Card or T-PAC
order	TPACA1	2	Ordering Information
reldoc	TPACA1	2	Part Numbers and Related Documents
second	TPACA1	3	System Hardware Configuration
abusio	TPACA1	4	A-BUS Inputs and Outputs
dataflo	TPACA1	5	Data Flow
hardwar	TPACA1	6	Hardware Description
ros	TPACA1	6	Read-Only Storage (ROS)
seq	TPACA1	7	Microprogram Sequencer (PN 6832982)
alu	TPACA1	8	Arithmetic Logic Unit (ALU) (PN 4745549)
fsu	TPACA1	8	Functional Storage Unit (FSU RAM) (PN 5120745)
lsr	TPACA1	9	Local Storage Register (LSR) (PN 8041469)
xreg	TPACA1	9	External Registers (PN 8494715)
xcvr	TPACA1	10	Front End Transceiver (PN 2405058)
clk	TPACA1	12	Clock Module (PN 2405060)
timer	TPACA1	14	16 Bit Timer Module (PN 6017383)
codform	TPACA1	14	Microcode
bitlow	TPACA1	14	ROS Configuration Bits(0-17)
bithigh	TPACA1	15	ROS Configuration Bits(18-48)
format	TPACA1	16	ROS Microinstruction Word Format
table	TPACA1	16	ROS Microinstruction Word, Bit Definitions
caution	TPACA1	19	Microcode Instruction Cautions
ctc	TPACA1	19	Command Tag Combination (CTC)
xregasn	TPACA1	20	External Register Assignments
xradr	TPACA1	20	External Register Addressing
io	TPACA1	24	User Interface
led	TPACA1	24	Led's
dip	TPACA1	25	Dip Switch
tap	TPACA1	26	10 Position Tap Switch
twoway	TPACA1	27	Single Pole Double Throw Switches
tc	TPACA1	29	Appendix A. Test Connectors

Figure ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Figure References</u>
sys21vl	TPACA1	3	1: 6
abus	TPACA1	4	2: 4
dataflo	TPACA1	6	3: 5
seqenc	TPACA1	8	4: 7, 14, 14, 16
xreg	TPACA1	9	5: 9
rxdata	TPACA1	10	6: 10
txdata	TPACA1	11	7: 7
clk	TPACA1	12	8: 12
cycle	TPACA1	12	9: 7, 12, 13
mentim	TPACA1	13	10: 12, 12
highbit	TPACA1	15	11: 14, 16
rosword	TPACA1	16	12: 14
xrgr0	TPACA1	21	13: 13
xrgr1	TPACA1	22	14: 14
xrgr2	TPACA1	23	15: 15
25pin	TPACA1	24	16: 16
9pin	TPACA1	24	17: 1

jj1	TPACA1	29	18:
jj2	TPACA1	29	19:
jj3	TPACA1	30	20:
jj4	TPACA1	30	21:

Imbed Trace

EPROM4-A HARDWARE FUNCTIONAL SPECIFICATION

Control Systems Engineering

**IBM Corporation
General Technology Division
D-035 B-300-47A
East Fishkill
tel 533-7654**

IBM Internal Use Only

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

TABLE OF CONTENTS

Eprom4-A Functional Specification	1
General Information	1
Part Number Information	1
Features	1
Card Overview	1
Function	3
A-BUS Interface Definition	3
A-BUS Inputs and Outputs	4
User Interface	4
LED's	5
Hardware Programming	5
24 Volt Power Connector	6
CTC Definitions	6
Timing Diagrams	7
Software Functional Specification	9
How to Burn	10
Appendix A. Register Bit Definitions	13

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

LIST OF ILLUSTRATIONS

Figure 1.	Second Level Diagram	2
Figure 2.	EPROM-A ABUS 2nd LEVEL	3
Figure 3.	Jumper Configuration	5
Figure 4.	Power Connector Placement	6
Figure 5.	Eprom4-A Read Timing Diagram	7
Figure 6.	Eprom-A Write Timing Diagram	7
Figure 7.	Configuration Screen	9
Figure 8.	Main Menu Screen	10
Figure 9.	Verify Eprom Option	11
Figure 10.	LOAD SCREEN	11
Figure 11.	EPROM DATA BITS (0-15)	13
Figure 12.	EPROM ADDRESS BITS (0-15)	14
Figure 13.	EPROM CONTROL REGISTER BITS (0-7)	14

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

EPROM4-A FUNCTIONAL SPECIFICATION

GENERAL INFORMATION

This specification explains the hardware design and function intended for the Eprom4-A card. It will bring forth any limitations and describe all hardware programming pins or switches necessary for proper operation.

It is assumed that the user is familiar with the DIS BUS and the protocol used to communicate to/from cards on the DIS BUS. Also, The user must be familiar with the ARIEL system and EDX-J for programming.

Part Number Information

Assembly	--> 71X8902
Raw Card	--> 71X8903
Schematic	--> 71X8904
EC	--> A37493Q

FEATURES

1. Can Burn
 - 2764A
 - 27128A
 - 27256A
2. Max burn (32K x 16 bits) x 4 groups
3. 1 Group is hardware programmable as read only
4. 24 Volts is regulated to desired burn voltage 12.5/5.0 volts.
5. 8.5 Volts is regulated to desired Vcc voltage 6.0/5.0 volts.
6. Implemented as a Macro-Function Card
 - MACRO-ID = 'FE' hex

CARD OVERVIEW

The EPROM4-A card will be used to burn Eproms. This function will be implemented through the DIS Bus. Thus Eprom4-A will be another macro-function card and will conform to the DIS Standards. It will be able to burn 2764A, 27128A, and 27256, The card has 4 groups of eproms (2/group) all of which are programmable, and the fourth group can be used either as a master (read only) or can be programmable.

IBM Internal Use Only

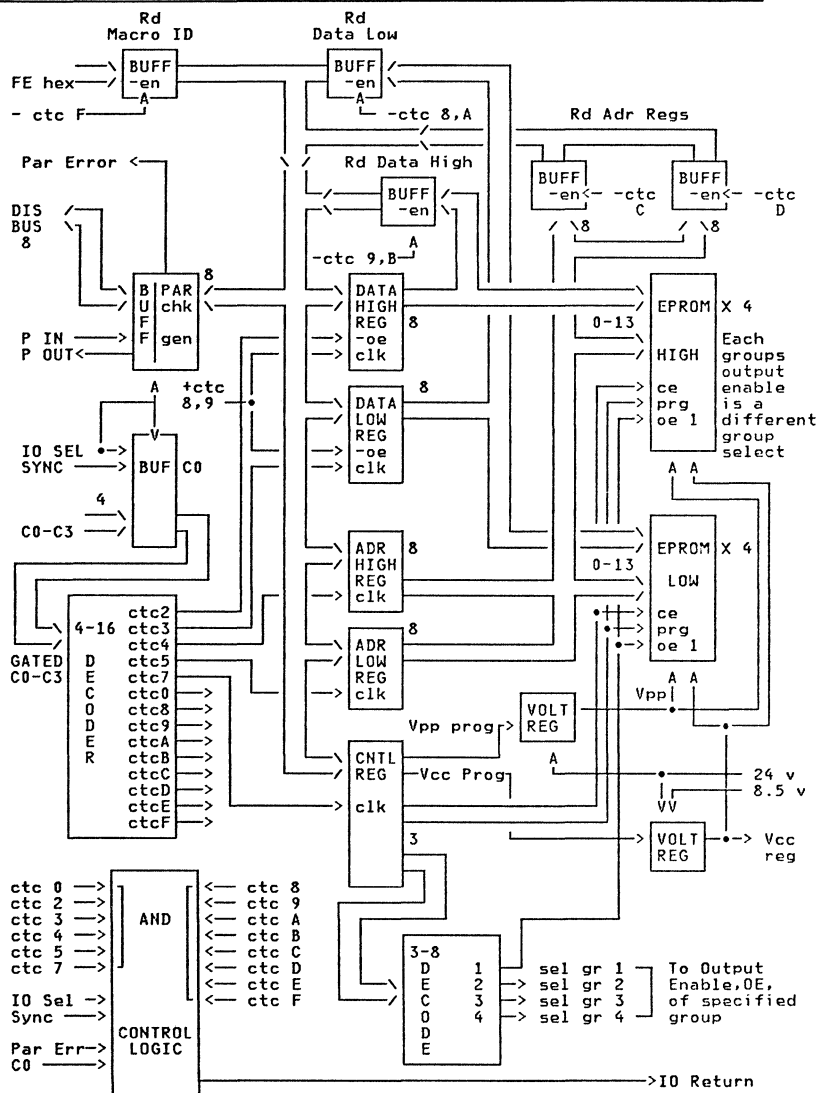


Figure 1. Second Level Diagram: There are four groups of eeproms each wired as shown except group 4, it can be jumpered so that it is not programmable.

FUNCTION

Data is transferred to/from Eprom4-A through the DIS Bus (IO Bus). This bus is buffered so that each input to Eprom4-A has only one load. Parity is checked on DIS writes to Eprom4-A and generated for DIS reads. CTC's 2, 3 will latch Eprom data. CTC's 4, 5 will latch the Eprom address. And CTC 7 will latch the Control Signals. CTC's 8 thru F are used to read Eprom data, Address reg, Data reg, Control reg, and Macrofunction ID. CTC 0 is used to generate the select to Eprom4-A via the Communications Adapter, but Return is generated from Eprom4-A, not the Communications Adapter.

The output enable lines of each eeprom group are decoded from 3 signals from the control register, called group selects. There are also 2 voltage regulators on card. The Vpp program signal controls Vpp to be 12.5 volts or + 5v. The Vcc program signal controls Vcc to be + 6v or + 5v. These are controlled through the control register also.

The Eprom Data regs are disabled (3-st) during a ctc 8 or 9, read Eprom4-A Data. This is to ensure no data collision on the eeprom data bus high or low. The Eprom address and control regs are always enabled. This is necessary for control of the read/write commands. Also, all the DIS writes are CTC 2-7 and DIS reads are CTC 8-F. Thus the C0 bit is used as the read/write line.

C0 Active (gnd) DIS read
C0 Inactive (+5v) DIS write

Programming and reading data from the Eproms is very simple. Eproms are programmed by latching data and address into their respective registers then controlling the burn sequence as described in INTEL's "Memory Components Handbook" 1985, via the control register's Chip Enable, Program, Vcc, Vpp, and the Output Enable signals.

A-BUS INTERFACE DEFINITION

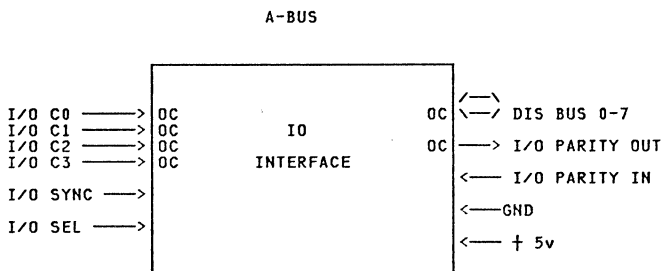


Figure 2. EPROM-A ABUS 2nd LEVEL: The DIS channel is an open-collector bus. The terminators for the OC signals will reside on the board.

IBM Internal Use Only

A-BUS Inputs and Outputs

The A-Bus will provide the path for data transfer to/from the Eprom4-A card. All control, Data Busses, Address, and Power will be provided from the A-Bus, except the Program Voltage (VPP).

The following is a list of all the A-Bus I/O on Eprom4-A and their definitions as a function of the Hazel-A card. These I/O are shown in Figure 2 on page 3.

- IO SYNC OUT I DIS Sync ; When active, signals data and CTC's are valid on the DIS Bus to the Eprom4-A card provided IO Select is active.
- IO RETURN IN O DIS Return ; When reading Eprom4-A, this output signals data is valid on the bus. When writing Eprom4-A, this output acknowledges the data has been received and is valid, (ie. good parity). It is enabled when Sync, and IO Sel are active during a DIS Read, and Sync, IO Sel, and parity is valid during a DIS Write provided a valid ctc has been sent.
- IO SELECT I DIS LSA Select ; This input Enables Eprom4-A. When active, it will allow the DIS Bus, C0-C3, Sync, and Return through it's buffers.
- C0,C1,C2,C3 I Command Tag Combination ; These inputs are used as control signals to Eprom4-A. They are valid when IO Sel, and Sync are valid.
- IO D0-D7 I/O DIS Bus Bits 0-7 ; These bidirectional open-collector signals are the data bus to/from the Eprom4-A card. When reading Eprom4-A, data is valid before Return becomes active. When writing Eprom4-A, data is valid before Sync becomes active.
- IO DP IN O DIS Data Parity IN ; This output is generated to guarantee Negative Active ODD Parity across the DIS Bus (ie. an odd number of gnnds on the DIS Bus and IO DP IN)
- IO DP OUT I DIS Data Parity Out ; This input and the DIS Bus are checked to guarantee Negative Active ODD Parity during a DIS Write. If Parity is bad (ie. not an odd number of gnnds across the 9 bits), then return will not be generated.
- Voltages I There are 9 different voltage levels available on the Abus connector. They are;

Voltages		Pins	
1.	+ 5v	115, 117, 119	
2.	gnd	116, 118, 120	
3.	+ 8.5v	33	
4.	- 5v	71	not used
5.	+ 12v	48	not used
6.	- 12v	94	not used
7.	+ 15v	1	not used
8.	- 15v	89	not used
9.	Agnd	2	not used

Note: Agnd is Analog ground and should be kept separate from gnd (digital ground).

USER INTERFACE

This section will detail the function of the LED's, hardware jumpers, and external power supply by section.

IBM Internal Use Only

LED's

There are 5 led's on Eprom4-A. Four are used for the programming voltages and one signals when the card is selected on the DIS Channel.

LED 1 --> I/O Select (LSA)
LED 2 --> Signals Vpp is at 12.5 volts
LED 3 --> Signals Vcc is at 6.0 volts
LED 4 --> Signals Vpp is inoperational
LED 5 --> Signals Vcc is inoperational

Hardware Programming

There are three jumpers on Eprom4-A. These hardware program pins are implemented to allow for changes and versatility in burning different sizes and types of eproms. See Figure 3 for position information.

Group 1 This jumper is used to select the 256K density eproms

Position 1 --> burn 2764A, 27128A
Position 2 --> burn 27256

Group 2 This jumper multiplexes Chip Enable with Program

Position 1 --> Pin 20 of Eprom socket is Chip Enable
Position 2 --> Pin 20 of Eprom socket is Program

Group 3 This jumper selects Eprom group 4 as burn or not to burn

Position 1 --> Eprom Group 4 is programmable
Position 2 --> Eprom Group 4 is READ ONLY

Note: Group 2 Jumper must be in Position 1 for the Eprom burn Software (BURN2) to function properly.

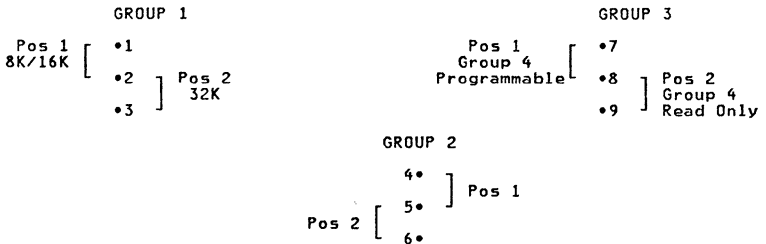


Figure 3. Jumper Configuration: Jumpers 1-3 are located near the top back of the card, near the 24 volt power input connector. Jumpers 4-6 are located below Eprom Group 1. Jumpers 7-9 are located between Eprom Groups 3 and 4 near the top of the card.

IBM Internal Use Only

24 Volt Power Connector

The 24 volts to Eprom4-A is brought in through SPE1 and SPE2. These are right angle BERG headers located on top of Eprom4-A near the back. Both of these headers are wired the same and are inter-changeable. They are wired as shown in Figure 4;

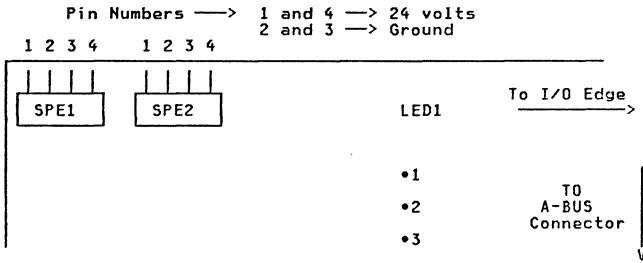


Figure 4. Power Connector Placement: There are two power connectors. If burning multiple cards, the 24 volt power can be jumpered from card to card.

CTC DEFINITIONS

CTC 0	IO Sel
CTC 2	Write Eprom Data High to Register
CTC 3	Write Eprom Data Low to Register
CTC 4	Write Eprom Address High to Register
CTC 5	Write Eprom Address Low to Register
CTC 7	Write Control Register
CTC 8	Read Eprom Data High
CTC 9	Read Eprom Data Low
CTC A	Read Data Reg High
CTC B	Read Data Reg Low
CTC C	Read Adr Reg High
CTC D	Read Adr Reg Low
CTC E	Read Control Reg
CTC F	Read Eprom Macrofunction ID
CTC 1	NOT USED
CTC 6	NOT USED

IBM Internal Use Only

TIMING DIAGRAMS

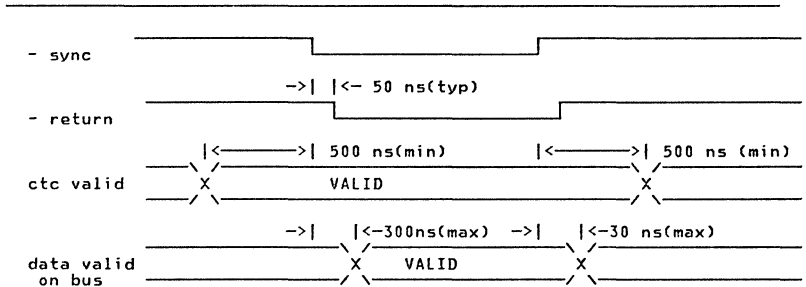


Figure 5. Eprom4-A Read Timing Diagram

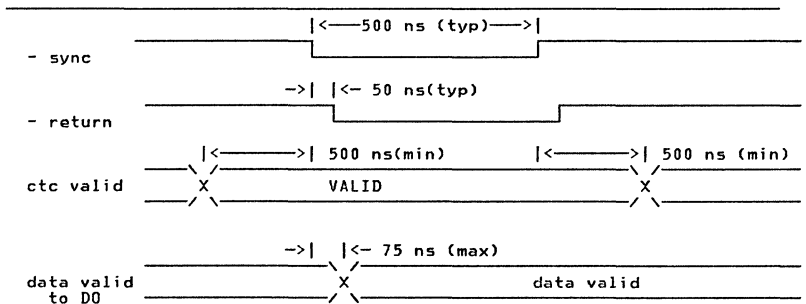


Figure 6. Eprom-A Write Timing Diagram: Parity is checked on a DIS write. If parity is bad then return is blocked to the processor.

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

SOFTWARE FUNCTIONAL SPECIFICATION

Burn2: Upon loading "Burn2" the user will be prompted to enter the configuration of Eproms to be burned (ie. 8/16/32 K eproms, 8/16 bits wide, and 1 or 2 cards). The number of cards (2 maximum) defines the total length of burn (ie. If 8K eproms and 1 card were selected then total burn length = 8k, but if 2 cards were selected instead then the total burn length = 16k). The maximum burn is limited by "Burn2" and is 64K x 16 using 32K Eproms.

Note: "Burn2" will scan all the I/O (each LSA) until the ID of the card/cards matches 'FE' hex, and uses the first 2 Eprom4-A cards found in the system. The code will display each card's Block and Macro address for verification.

After initializing the code with the type, width and length of the eproms being burned, the configuration will be displayed on the 3101 and the user will be prompted if the configuration is correct. See Figure 7 as an example.

```
YOU ARE BURNING 16K ON PART NUMBER 2764-A / 6231573 EPROMS

CARD 1    BLOCK = 2    MACRO = 3

      HIGH          LOW
      BYTE          BYTE
0000 *****          *****
      * 8K *        * 8K *
      * x  *        * x  *
      * 8  *        * 8  *
      *   *        *   *
1FFF *****          *****

CARD 2    BLOCK = 2    MACRO = 6

      HIGH          LOW
      BYTE          BYTE
2000 *****          *****
      * 8K *        * 8K *
      * x  *        * x  *
      * 8  *        * 8  *
      *   *        *   *
3FFF *****          *****

IS THIS CONFIGURATION CORRECT ?    Y/N

on
```

Figure 7. Configuration Screen: After selecting the type, width and length of eprom to use the configuration will be shown on the 3101. If an eprom card is not found an error will be detected and the user will be prompted for a new configuration.

After verifying the configuration the MAIN MENU will be displayed on the 3101. It is described in Figure 8 on page 10

IBM Internal Use Only

```
*****
*                               *
*             IBM               *
*          EPROM CARD TEST      *
*                               *
*          9/03/86-16.10        *
*          ID=00FE              *
* NOTE : ALL NUMBERING IN IBM NOTATION *
*       : BIT 0 = MSB           *
*       : BYTE 0 = MSB = HIGH BYTE *
*                               *
*****

ENTER : 1) INITIALIZE PROGRAM
        2) LOAD MEMORY
        3) BURN EPROM
        4) VERIFY BURN
        5) BLANK CHECK
        6) BURN + VERIFY  r--- last option selected
        7) EDIT MEMORY    v
        99) END PROGRAM   ( 0) __ <-- Enter option here or hit enter
```

Figure 8. Main Menu Screen: Enter the desired option. If no option is entered, the last option selected will be used. This is displayed in paranthesis across from the last option.

HOW TO BURN

Now The user has to select a sequence of options to burn the eproms. The first option is to load memory with data, then burn memory and then verify if required (verification should be used). The Blank check option can be used any time before burning the eproms.

Note: When burning, verifying, or blank checking eproms, the user must enter the groups being used. An example is shown in Figure 9 on page 11

IBM Internal Use Only

```

*****
*                               IBM                               *
*                               EPROM CARD TEST                   *
*                               *
*                               9/03/86-16.10                     *
*                               ID=00FE                           *
* NOTE : ALL NUMBERING IN IBM NOTATION                           *
*       : BIT 0 = MSB                                           *
*       : BYTE 0 = MSB = HIGH BYTE                               *
*****
ENTER : 1) INITIALIZE PROGRAM
        2) LOAD MEMORY
        3) BURN EPROM
        4) VERIFY BURN
        5) BLANK CHECK
        6) BURN + VERIFY      r--- last option selected
        7) EDIT MEMORY        v
        9) END PROGRAM      ( 0) 4  <-- Enter option here or hit enter

VERIFY EPROM
BLANK

ENTER GROUP (1234) ____ <-- Enter groups to be verified here

A
L-- defines the group/groups to be verified
    this designates that groups 1, 2, 3, and 4
    are going to be verified, unless new groups
    are entered.

```

Figure 9. Verify Eprom Option: The previous group/groups verified is shown in parenthesis. To change the groups just enter those required. They can be in any order and as few as one, but do not enter more than four and do not enter the same group twice. This is true for Verifying, Blank Checking, and Burning Eproms.

Another feature is while setting up to verify or blank check, the user is prompted for the LAST ADDRESS { }. When entered, "Burn2" on page 9 will only verify or blank check to that address. Also when burning, the last address is taken from the blank check LAST ADDRESS.

After blank checking the eproms, the user should load memory. When selected (option 2), the LOAD screen will be displayed on the 3101 screen as shown in Figure 10.

```
ENTER LOAD SOURCE :  1) EPROM
                   2) HOST
                   3) DISK
                   99) EXIT
```

Figure 10. LOAD SCREEN: The user has the option to load from EPROM (Group 4 only), DISK, or HOST.

Ariel Memory is loaded in two areas: Partition 1 and Partition 2. If 1 card is selected to burn (ie. 32K x 16 using 32K eproms), then the program or data to be burned will be loaded into Partition 1 of Ariel memory (Adr '8000' hex). If 2 cards are selected to burn (ie. 16K x 16 using 8K eproms), then the first 8K of Eprom data (Adr '0000' to 'FFFF' hex) will be loaded in Partition 1 starting at Address '8000' hex of Ariel memory. The second 8K of eprom data (Adr '2000' to '3FFF' hex) will be loaded into Partition 2 starting at Address '8000' hex of Ariel memory.

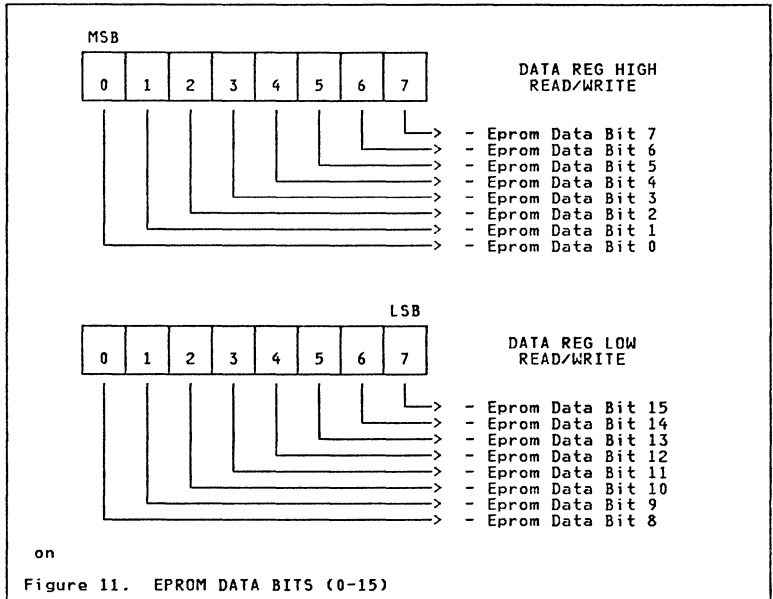
IBM Internal Use Only

After loading memory the user can EDIT MEMORY (option 7). Memory will be displayed starting at Address '0000' to the end of the program or data. It will not be displayed like Ariel Memory but exactly like the users listing or compiled data.

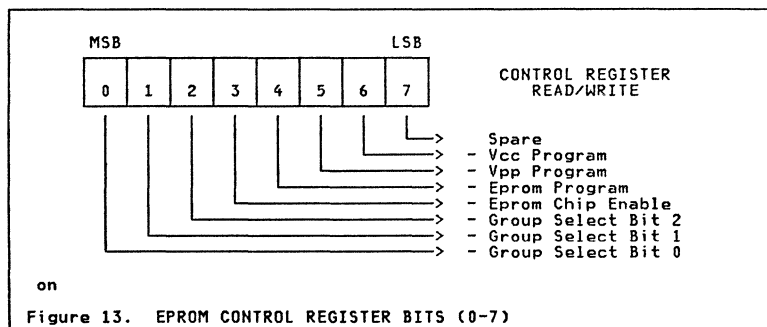
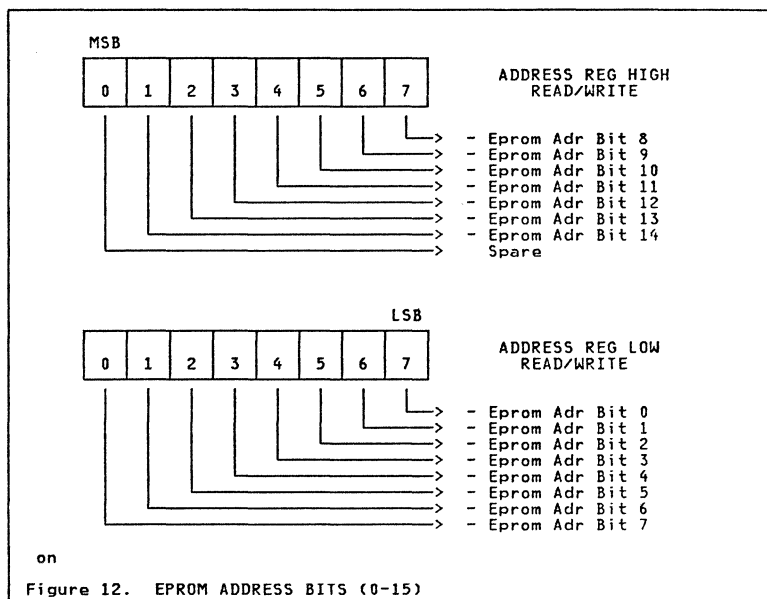
Now the user can BURN EPROM (option 3), VERIFY BURN (option 4), or BURN + VERIFY (option 6). These are self-explanatory functions. It should be noted that while burning or verifying, status data will be displayed on the 3101 screen. The user can monitor this to determine how the burn or verify is progressing.

An Attention Key is available at all times to end what ever option has been selected. It is 'ATTN EL'. When prompted the user will be asked to hit the 'ENTER' key or type in 'EN'. If the 'ENTER' key is selected, the burn will continue from where it left off, and no data will be lost. If 'EN' is typed in, the selected option will terminate and the main menu will be displayed.

APPENDIX A. REGISTER BIT DEFINITIONS



IBM Internal Use Only



Heading ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Heading References</u>
abusio	EPR0M4A	4	A-BUS Inputs and Outputs
pow	EPR0M4A	6	24 Volt Power Connector
code	EPR0M4A	9	Burn2 9, 9, 9, 11

Figure ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Figure References</u>
epdata	EPR0M4A	2	1:
habuss	EPR0M4A	3	2: 4
jump	EPR0M4A	5	3: 5
24volt	EPR0M4A	6	4: 6
eprrd	EPR0M4A	7	5:
eprrr	EPR0M4A	7	6:
conf	EPR0M4A	9	7: 9
main	EPR0M4A	10	8: 9
verif	EPR0M4A	11	9: 10
load	EPR0M4A	11	10: 11
databit	EPR0M4A	13	11:
adrbitt	EPR0M4A	14	12:
cntlbit	EPR0M4A	14	13:

Imbed Trace

```

DSMLBR529E 'KP' WOULD EXCEED MAXIMUM SIZE.
DSMMOM395I '.DSMEFIG' LINE 140: .hr afigrule left to right
DSMMOM397I '.DSMEFIG' WAS IMBEDDED AT LINE 122 OF 'EPROM4A'
DSMMOM397I 'EPROM4A' WAS IMBEDDED AT LINE 25 OF 'MAIN2'
DSMKDF474W FONT SAVE STACK EMPTY - DEFAULT ASSUMED.
DSMMOM395I '.DSMEFIG' LINE 160: .pf
DSMMOM397I '.DSMEFIG' WAS IMBEDDED AT LINE 122 OF 'EPROM4A'
DSMMOM397I 'EPROM4A' WAS IMBEDDED AT LINE 25 OF 'MAIN2'
DSMBEG323I STARTING SECOND PASS.
DSMLBR529E 'KP' WOULD EXCEED MAXIMUM SIZE.
DSMMOM395I '.DSMEFIG' LINE 140: .hr afigrule left to right
DSMMOM397I '.DSMEFIG' WAS IMBEDDED AT LINE 122 OF 'EPROM4A'
DSMMOM397I 'EPROM4A' WAS IMBEDDED AT LINE 25 OF 'MAIN2'
DSMKDF474W FONT SAVE STACK EMPTY - DEFAULT ASSUMED.
DSMMOM395I '.DSMEFIG' LINE 160: .pf
DSMMOM397I '.DSMEFIG' WAS IMBEDDED AT LINE 122 OF 'EPROM4A'
DSMMOM397I 'EPROM4A' WAS IMBEDDED AT LINE 25 OF 'MAIN2'

```

Communication Sub-System Functional Specification

Control Systems Engineering

**IBM Corporation
General Technology Division
D-035 B-300-47A
East Fishkill
tel 533-7654**

IBM Internal Use Only

IBM Internal Use Only

THIS PAGE IS INTENTIONALLY LEFT BLANK

IBM Internal Use Only

TABLE OF CONTENTS

Related Documents	1
Architecture	3
ARIEL DSC	4
ATTACHED PROCESSOR	5
INTRODUCTION	5
Cycle Steal	5
AP/hazela	5
TECHNOLOGIES	5
PACKAGE	5
COMPONENTS	5
LOGIC	5
MEMORY	5
OTHER COMPONENTS	6
CONNECTORS	6
ORGANIZATION	6
DATA FLOW	6
REGISTER DEFINITION	6
EXTERNAL REGISTER INTERFACE	8
OVERVIEW	8
ADDRESSING	8
DATA INTERFACE	9
EXTERNAL REGISTERS FOR HAZELA INTERFACE	9
CONTROL LINES	11
INTERRUPT STRUCTURE	13
GENERAL	13
THEORY OF OPERATION	13
INTERRUPT HARDWARE DESCRIPTION	13
OVERVIEW	13
INTERRUPT MODULE	14
INTERRUPT LEVEL ASSIGNMENTS	16
INTERRUPT FROM hazela TO THE AP	17
hazela INTERRUPTS	17
PROGRAMMING CONSIDERATIONS	18
hazela CYCLE STEAL INTERFACE	18
OVERVIEW	18
PROGRAMMING CONSIDERATIONS	19
EPROM PROGRAMMING	19
INSTRUCTION SET DESCRIPTION	19
CONTROL STORAGE OP 1 (CS1)	20
CONTROL STORAGE OP 1 hazela (CS1H)	21
MEMORY TIMING	21
INSTRUCTION TIMING	22
ERROR CHECKING	23
GENERAL	23
JIB' DETECTED CHECKS	23
SUNSET MEMORY SPECIAL CONSIDERATIONS	23
CLOCK	25
JIB' TIMING	25
DIAGNOSTICS	27
External Register Names and Bit Assignments	29

IBM Internal Use Only

THIS PAGE IS INTENTIONALLY LEFT BLANK

IBM Internal Use Only

LIST OF ILLUSTRATIONS

Figure 1.	ARIEL DSS Logic Block Structure	3
Figure 2.	AP structure	7
Figure 3.	AP Data Flow	8
Figure 4.	AP External Register (XR) Addressing	9
Figure 5.	AP/hazela Memory Interface	10
Figure 6.	AP External Register Assignment (stack 0)	12
Figure 7.	AP Interrupt Module (8691943)	16
Figure 8.	AP Interrupt Level Assignments	17
Figure 9.	AP Cycle Steal Timing	19
Figure 10.	Clock Module Timings	25
Figure 11.	External Register Interface Timing	26
Figure 12.	AP LED DIAGNOSTICS	27

IBM Internal Use Only

THIS PAGE IS INTENTIONALLY LEFT BLANK

IBM Internal Use Only

RELATED DOCUMENTS

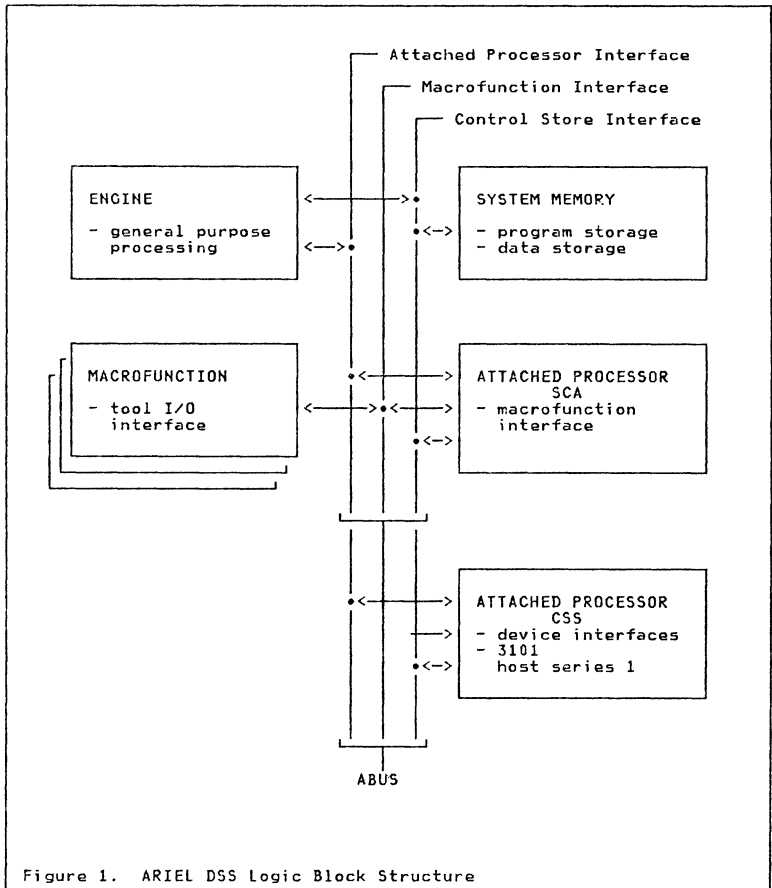
Related documents are:

- JIB' Functional Specification Microprocessor Development Group, Los Gatos, Calif.
- EDX/J Language and Supervisor Description, september 1986
- Functional Specification for the Memory Mapped Programmable Communication Adapter, doc. # AFS0027-01-RAL, by F.A. Newlin III
- Distributed Sensor Subsystem IV Functional Spec E45-1289-0 PN 7834982

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

ARCHITECTURE



The block diagram in Figure 1 is an illustration of the physical structure of an ARIEL DSS Logic Block showing how the system bus (ABUS) connects the four different types of Function Cards - Engine, System Control Store, Attached Processor, Macrofunction - which comprise the ARIEL DSS architecture. The Engine (ENG) card provides general-purpose processing capability within the Logic Block. The System Memory card provides data and program storage capability for the ENG. The Attached Processor (AP) cards act as slaves to the ENG providing it with intelligent interfaces to other computers, consoles, or other Logic Blocks. The Macrofunction cards provide the means of attaching to a tool's sensor-based I/O points.

IBM Internal Use Only

ARIEL DSC

The Distributed System Controller (DSC) is a general purpose controller with interfaces to a host computer, a system console, and an I/O channel. The host computer interface provides bidirectional communications to a computer of higher intelligence allowing the DSC to connect to a network of computers in an automation system with distributed intelligence. The system console interface provides a communication path to the operator. And the I/O channel interface provides attachment to a sub-network of lower intelligence I/O concentrators.

A Logic Block configured as a DSC-IV, provides a compatible replacement to existing DSS-IV hardware for users who do not require local disk support. The ARIEL DSC-IV is comprised of:

1. An ENG card for general processing of control algorithms.
2. A System Memory card for program/data storage.
3. An AP/CSS card with an RS-232C interface for short-medium distance communications (e.g. console) and an optically isolated interface for medium-long distance communications (e.g. host) CSS card.

IBM Internal Use Only

ATTACHED PROCESSOR

INTRODUCTION

The Attached Processor (AP) is a stored program I/O processor for use with the HAZELA processor in the East Fishkill Distributed System Controller. (ARIEL)

The intent of the AP is to offload I/O control from the hazela engine for those devices which require many cycles. These functions include host (series 1) and terminal (3101) communications.

The AP is a single card containing a processor, ROS and RAM memory, and interfaces to the hazela cycle steal bus. The card also has top card connectors for the COMMUNICATIONS SUB SYSTEM(CSS) interface, local control store address and data buses.

Cycle Steal

AP/hazela

The hazela cycle steal interface connects the AP and hazela as a tightly coupled (i.e. memory coupled) processor.

The AP includes, for diagnostic purposes, four LED's mounted on the card which, under program control, may be used to indicate the state of the AP. The LEDs are read as a hex value with the LSB being the one closest to the front aperture plate.

TECHNOLOGIES

PACKAGE

The AP is packaged on a single card with top-edge connectors. The internal structure is four signal planes wired two lines per channel on the planes. A unique internal voltage plane provides module power. The card requires supplies of +5, +8.5, and -5 Volts. The card assembly is P/N 6912781, schematic P/N is 6912783.

COMPONENTS

LOGIC

The AP uses the JIB' microprocessor module, a 28mm module with 116 pins, 11 by 11 pin array. The tristate drive for the hazela cycle steal bus is provided by MS-639 COMPASS drivers. All other logic on the card is implemented in GOLF (MS-438) technology, and in VTL.

MEHCRY

JIB' memory is implemented in two 32Kx8 Vendor EPROM's (INTEL 27256) and one 32Kx18 SUNSET II module (5122163). Memory is accessed as 18 bits wide (16 data bits and 2 parity bits). Parity is not generated by the EPROM's,

IBM Internal Use Only

so external VTL parity generators are used to prevent JIB from generating a parity error when accessing them. JIB' cycle time will be dependent on the cycle time of the memory being accessed, causing the JIB' cycle not to be a fixed length. It is intended that instructions be stored in EPROM, and that the SUNSET II be used for data buffering. It is possible, but not recommended, that instructions be stored in, and executed from the SUNSET II. Cycle time is 550 ns for control store and 333 ns for registers.

The user should be aware that the Soft Error Rate of the SUNSET II technology is non-zero. It is therefore necessary that the software allow for occasional soft parity errors in RAM. If the error occurs in an instruction stored in SUNSET II, recovery may be impossible.

The substrate voltage for the SUNSET is provided by a voltage divider from the -5 volt supply. SUNSET requires -1.5 Volts for the substrate.

The EPROM's occupy JIB' memory space between locations X'0000' and X'7FFF'. The SUNSET II memory is in locations X'8000' to X'FFFF'.

OTHER COMPONENTS

The oscillator is a 20 MHz crystal controlled VTL compatible circuit, vendor supplied, packaged in a TO-8 can.

The +1.7 Volt supply for the JIB' module is provided by an on-card regulator using a ZIRCON technology regulator (5616162.) and discrete components. The output is temperature compensated.

Five Light Emitting Diodes (LED's) are provided as indicators. four for diagnostics and one for run error, which can be seen from the front aperture plate.

Decoupling capacitors, resistors, transistors, R-packs, and program pins are IBM standard items.

CONNECTORS

The board connector of the AP card provides the power on reset, hazela cycle steal interface and the operating voltages.

The top card connectors contain the interface for the CSS, along with three test connectors for local control store address, data and clocking.

ORGANIZATION

DATA FLOW

The AP structure is shown in Figure 3 on page 8, and the AP data flow in Figure 2 on page 7. For further information on the JIB', refer to the JIB' functional specification.

REGISTER DEFINITION

The JIB' microprocessor is capable of addressing 128 Local Storage Registers (LSR's). These are contained in one SAMANTHA module (2041469). one class of instructions addresses registers as one byte fields, another class of instructions treats registers as even/odd pairs (halfwords). Therefore, the data path to and from the local storage registers and JIB' is organized as an 18 bit bus (16 bits of data and 2 bits of parity).

IBM Internal Use Only

Register data out busses from JIB' (DO, DE BUS OUT) provide the outgoing data to the LSR's and External Registers (XR's). The DE BUS OUT provides the data path to the even addressed LSR's (LSB of register address is zero). DO BUS OUT provides the data path to the odd addressed LSR's and the external registers(XR's).

Incoming data from LSR's and XR's to JIB' is carried on data bus in (DE, DO BUS IN). The DE BUS IN provides the data path for the even addressed LSR's. The DO BUS IN provides the data path for odd addressed LSR's, and all XR's.

Addressing and data bus steering for LSR's and XR's is performed by the JIB' microprocessor module through appropriate address and control lines.

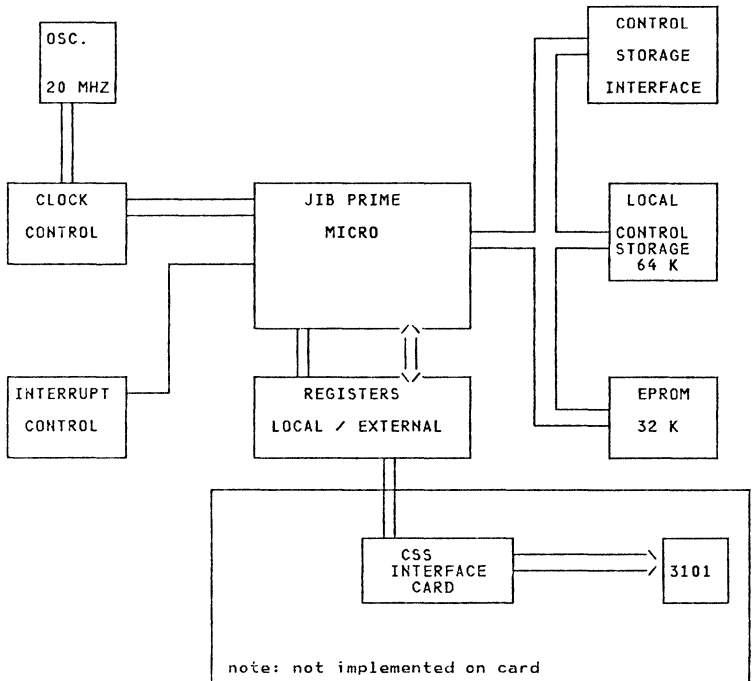


Figure 2. AP structure

IBM Internal Use Only

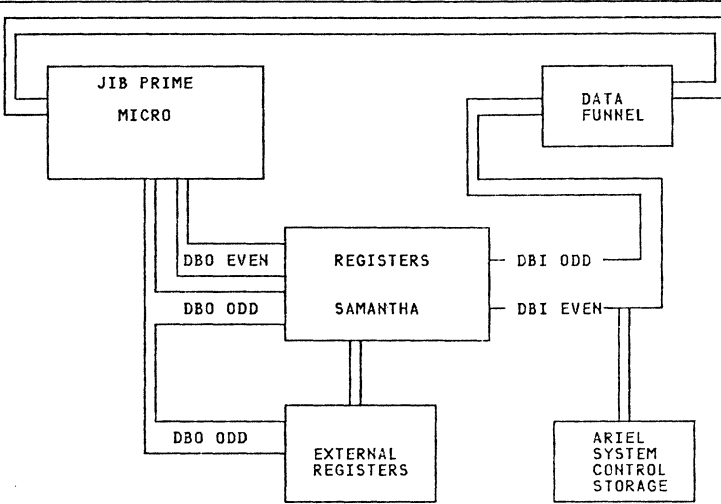


Figure 3. AP Data Flow

EXTERNAL REGISTER INTERFACE

OVERVIEW

The XR interface is the primary interface for the AP to control an attached device. It behaves in a similar manner to the standard JIB' XR interface.

XR's are implemented on the AP card for control of the interrupt structure, device selection, and the hazela cycle steal interface. JIB' transfers data to and from the XR's using the odd half of the busses used for the LSR's. Odd parity is maintained on these busses, but parity may be ignored if the Inhibit DE/DO Bus Check line to JIB' is held high. This may be the case if the attached device does not generate parity. A bidirectional bus to and from the XR's is generated on the AP card to minimize the number of card and module pins required by the XR data busses.

ADDRESSING

The addressing of the XR's is provided on eight lines. All address lines are negative active. Bits 0-2 will take on values from 1 to 7. Bits 3-7 will take on values from X'00' to X'1B'. This provides for a total of 196 addresses. The lines may take on other values if the XR interface is not selected.

IBM Internal Use Only

DATA INTERFACE

The data interface is a 9 bit bidirectional tristate bus, negative active. Odd parity must be provided by the attached device when transfer is from the attached device to the AP. The AP will provide odd parity on data outbound to the attached device.

Figure 6 on page 12 shows the address and function of all XR's implemented on the AP card.

EXTERNAL REGISTER ADDRESS SPACE EXTENSION: Since a device to be controlled by the AP may need many XR's, XR X'1C' is used to extend the XR address space. This is done as follows:

Bits 1,2 and 3 of XR X'1C' are called the XR Stack number. XR's X'1C', X'1D', X'1E', and X'1F' are common to all XR stacks, and are implemented on the AP card. XR's X'00' to X'1B' have the address extended by the XR stack number, providing a total of $8 \times 28 + 4$ or 228 individually addressable XR's. Of these, 7×28 , or 196 are available for device control, as XR stack 0 is reserved for XR's on the AP card itself. XR's on the attached card are selected only when the XR address is below X'1C' and the stack number is not 0.

The XR addressing is shown in Figure 4.

EXTERNAL REGISTER PARITY CONTROL: Bit 0 of XR X'1C' controls parity checking on the XR data bus into JIB'. If this bit is B'1', the DE/D0 check in JIB' is disabled. This bit is initially set to B'0'.

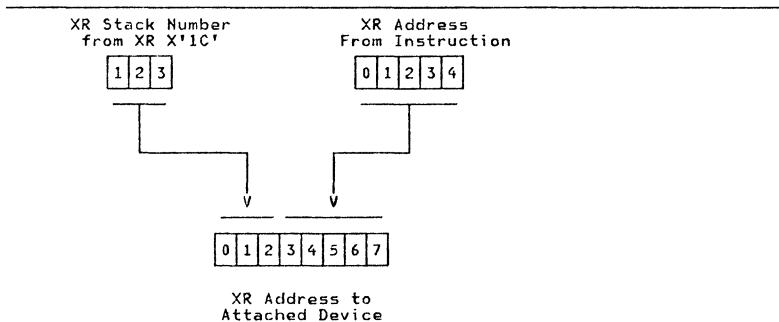


Figure 4. AP External Register (XR) Addressing

EXTERNAL REGISTERS FOR HAZELA INTERFACE

XR X'1B', and XR X'1C', are used in the control of the hazela cycle steal interface.

XR X'1B' contents are as follows:

- Bit 0
This bit controls the hazela reset. It is set to B'1' by POR, which resets the hazela engine. The program must not set this bit to 0 until the AP is ready to accept commands from hazela. The program must not set this bit to 1 once it has been set to 0, as this will reset the hazela engine, causing it to start execution at location X'00000'.
- Bits 1, 2, and 3

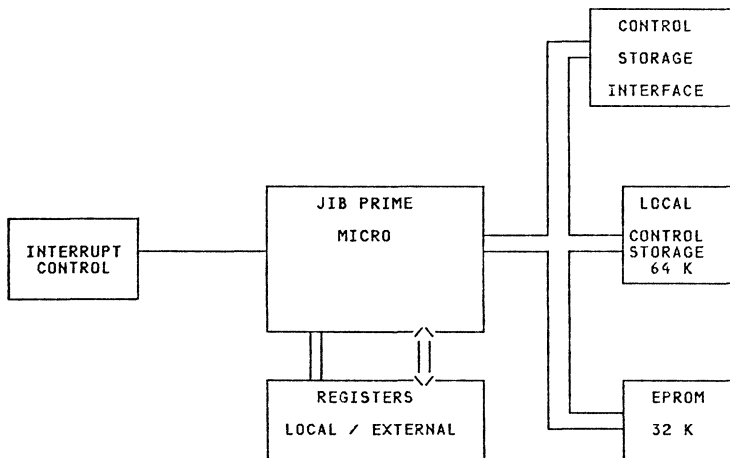
IBM Internal Use Only

These bits are not implemented, and are always B'0'.

- Bit 4
This bit indicates a parity check detected on the XR bus from JIB'.
- Bits 5, 6, and 7
These bits contain the low order three bits of the trigger address assigned to this AP. It is recommended that the program use these bits, rather than hard coding the trigger address to provide flexibility in trigger address assignment, and to allow more than one AP of a given type to be attached to the hazela. These bits are read only.

XR X'1C' contents are as follows:

- Bits 1, 2, and 3
These bits are used for other functions. See "EXTERNAL REGISTER ADDRESS SPACE EXTENSION" on page 9.
- Bit 4
This bit controls the interrupt request to hazela. If it is B'1', an interrupt request is generated.
- Bit 5
This bit is diagnostic select enable, it is used in conjunction with the ARIEL EPROM burn system.
- Bit 6
This bit is chip enable and is used for a EPROM burn card and enables the modules to be read or written.
- Bit 7
This bit is program enable and is used for a EPROM burn card and enables the modules to be written(burned).



IPL Device Switch

Figure 5. AP/hazela Memory Interface

IBM Internal Use Only

CONTROL LINES

Control lines provide for selection, data gating, timing control, and directional control on the data bus. The timing is shown in Figure 11 on page 26.

- DEVICE XR SELECT
The SELECT line is negative active, and indicates that a valid address is present on the address lines.
- SET REG GATE OUT
This line controls the gating of the tristate drivers. When the line is negative, data is transferred from the AP to the attached device.
- REG LOAD COMMAND
This line controls the gating of data into the attached device. It is negative (i.e. active), only when SET REG GATE is negative. The attached device uses this line to gate data from the data bus into the XR's.
- AP CLOCK HOLD
The XR interface provides a HOLD line, which, if held at the down level, will stop the JIB' clocks. This is provided to enable slower technologies (such as EMERALD) to be attached to the AP. When the AP reads from the XR interface, the data must be available at the AP XR DATA lines 80 nSec after SELECT goes active. If the device cannot meet this requirement, then fast logic must be used to generate the HOLD line. The attached device must provide a fast register for data outbound from the AP, as this line is not sensed in the AP at that time. The HOLD line may also be used under these conditions, with the understanding that the attached device must latch up the XR DATA and XR ADDRESS lines, as these will change before the HOLD line is sensed.

The external hardware must not pull this line down unless a stop is required to the JIB' clock, since this line stops JIB' unless the JIB' is actively accessing AP or hazela memory, or if a cycle steal is in process.

The JIB' clock transitions occur at the rising edge of - 20 MHz JIB Clock. The transition of the JIB Clock following the activation of HOLD is delayed until HOLD is made inactive. The next JIB clock transition occurs at the next rising edge of - 20 MHz JIB Clock after HOLD goes inactive.
- 20 MHZ CLOCK DELAYED
This line is a continuously running clock with a 50% duty cycle which may be used on the attached device for timing. The rising edges are synchronized to the transitions of the JIB' TA, TB, TC, and TD clocks. HOLD must change states while this line is at the up level to prevent metastability in the AP clock latches.
- XR DATA
These lines carry the data between the AP and the attached device.
- XR ADDRESS
These lines carry the XR address as contained in the JIB' instruction.
- XR STACK SELECT
These lines carry the XR stack number from XR X'1C'.

IBM Internal Use Only

XR Address	XR Name	
1F	Local Store Page Register (LSPR)	
1E	Interrupt Request Register	
1D	Interrupt Mask Register	
1C	XR Stack / C/S Page / hazela Interrupt and Inhibit DE/DO Check	
	XR Stack 0	XR Stack 1-7
1B	Trigger Address IPL flag and hazela Reset	IMPLEMENTED By APPLICATION (ON ATTACHED CARD)
1A to 08	NOT IMPLEMENTED	
07	Cycle Steal Page Value P0 Write Enable Run Error	
06	State Indicator Value	
05	Program Switches	
04	Error Status	
03 to 00	NOT IMPLEMENTED	

Figure 6. AP External Register Assignment (stack 0)

IBM Internal Use Only

INTERRUPT STRUCTURE

GENERAL

AP interrupt facilities include the following capabilities:

- 8 Prioritized interrupt levels.
- Automatic hardware controlled PSW swap.
- PSW storage are for inactive levels stored in LSR's
- Automatic latching of external interrupts

THEORY OF OPERATION

JIB' samples its interrupt request line at the T7 time of the JIB' cycle. If the line is active, JIB' enters a PSW swap sequence as described in the JIB' section of this document. The PSW swap time, i.e., the time delay between execution of the last instruction at the old level until the fetch of the first instruction at the new level, is 2 machine cycles, or 1100 usec.

Interrupts may be generated by the external hardware (an external interrupt), or by JIB' instructions (a programmed interrupt). The facilities for doing these functions are described in the section on interrupt hardware.

INTERRUPT HARDWARE DESCRIPTION

OVERVIEW

The interrupt hardware provides the masking, priority and interrupt request facilities for 8 prioritized interrupts. The lowest priority interrupt is assumed to be always active and always masked on, leaving 7 interrupts available to JIB' and the attached device. When one of the 7 interrupt requests becomes active and is not masked off and is on a higher level than the currently executing level, the following sequence occurs:

The priority network generates a three bit code corresponding to the highest outstanding interrupt request level. The compare network, recognizing a mismatch between the encoded new level and current level, generates a level swap request to the JIB' microprocessor. The microprocessor will use its Current Interrupt Level register to generate the address to store the current PSW, and will use the interrupt level lines to generate the address to fetch the new PSW. At the completion of the PSW swap the microprocessor will set its Current Interrupt Level register from the interrupt level lines and will raise Interrupt Response. This resets the Interrupt Request Latch in the external hardware. One instruction in the new interrupt level is guaranteed to be executed before another interrupt level switch can occur, even if a higher level interrupt becomes active.

The interrupt hardware sets the bit corresponding to an external interrupt in the Interrupt Register. The register will hold this interrupt level active until it is reset by the JIB' program even though the original interrupt request line is deactivated. The last instruction of the interrupt service routine must turn off the corresponding bit for its level in the Interrupt Register. This will cause a PSW swap back to the next highest active level.

Note: Due to timing constraints in the external hardware, JIB' may execute two more instructions before swapping levels. The level swap will occur

IBM Internal Use Only

even if the external interrupt request for the current level remains active. The external interrupt request will, however, not be lost.

Programmers should note that resetting a bit in the Interrupt Register other than the one corresponding to the current interrupt level may cause that interrupt to be lost. This will occur if the external interrupt is no longer active.

To prevent problems within the mask and interrupt registers, these registers are assigned to XR addresses within the group which inhibits detection of interrupts. This enables the program to manipulate these registers without causing invalid or erroneous interrupts.

INTERRUPT MODULE

The Interrupt handler is implemented on one GOLF Module (8691943) (Figure 7 on page 16) which includes:

- The Interrupt Request Register

This register indicates the levels which have outstanding interrupts. A bit in this register containing B'1' indicates that an interrupt is pending on the corresponding level. The bit assignments are shown in external register X'1E'. Bits may be set by either JIB commands to XR X'1E', or by external interrupts. Bits may be reset only by JIB commands.

Bits in this register may only be set by external interrupts if no interrupt request to JIB' is outstanding. External interrupts are sampled at JIB' T6 time and latched at JIB' T2 time, so that even if the external interrupt drops, the interrupt bit corresponding to that interrupt will remain on.

Note: Bit 7 of this register is not implemented, and will always be B'0'.

POR resets this register to X'80' (Level 0 interrupt), forcing the next level lines to X'0', thereby ensuring the JIB' starts execution in level 0.

- The Interrupt Mask Register

This register is used to mask the contents of the Interrupt Request Register. Only those interrupt levels which have the corresponding bits set to B'1' in this register may cause an interrupt. Bit 7 of this register is not used to mask interrupt level 7 (as this level is always active), but is used as a 'Master Mask'. If bit 7 is set to B'0' no interrupts will be generated to JIB, unless the interrupt is to level 0. Interrupts to level 0 may only be masked by setting bit 0 of this register to B'0'.

JIB' may set or reset this register via any XR instruction to XR X'1D'. Bit assignments for this register are shown in external register and bit assignments.

POR resets this register to X'80', allowing the level 0 interrupt to appear on the next level lines.

- The Interrupt Priority Tree

This circuit determines the highest outstanding interrupt which has not been masked off. When this value is different from the Next Level Register, an interrupt is generated to JIB, and the Next Level Register is updated.

- The Next Level Register

This register holds the next interrupt level on which JIB' is to operate. The register is not accessible from JIB' instructions, and is updated by the hardware at JIB' T2 time. This is required because the next level to JIB' must be stable before T7 time. POR Resets this

IBM Internal Use Only

register to X'0'. If the Interrupt Request Latch is on, updating the Next Level Register is delayed until the Interrupt Request is dropped.

- The Level Comparator

This circuit compares the contents of the Next Level Register and the output of the Interrupt Priority Tree. If the two values are unequal, the Interrupt Request Latch is set at JIB' T2 time, simultaneously with the Next Level Register being updated.

Interrupt Timing: Setting or resetting a bit in either the Interrupt Request Register or the Interrupt Mask Register (other than the Master Mask) from the instruction stream will take effect (causing a PSW swap), at the end of the second cycle following the instruction which caused the set/reset. This is caused by the internal clocking of the Interrupt GOLF module. This implies that the next instruction will always be executed and that the instruction following that may execute before the PSW swap occurs.

Setting or resetting the Master Mask will take effect during the instruction which set/reset the Master Mask.

Note: One more instruction will always be executed in the current level before a PSW swap (if any) can occur because JIB' does not sample the interrupt request during the instruction which manipulates the Master Mask.

When an external interrupt goes active, JIB' will not swap PSW's for at least 2 cycles, since it takes this many cycles to propagate the interrupt through to the Next Level register and the Interrupt Latch.

Machine Checks Due to SUNSET II Parity Checks: If a Machine Check occurs due to a SUNSET II parity error, the JIB' will swap to interrupt level 0 as long as it is not masked off. Since a parity check from SUNSET may be a soft error, it may be possible to recover. However, the data in the location with the parity error is lost.

IBM Internal Use Only

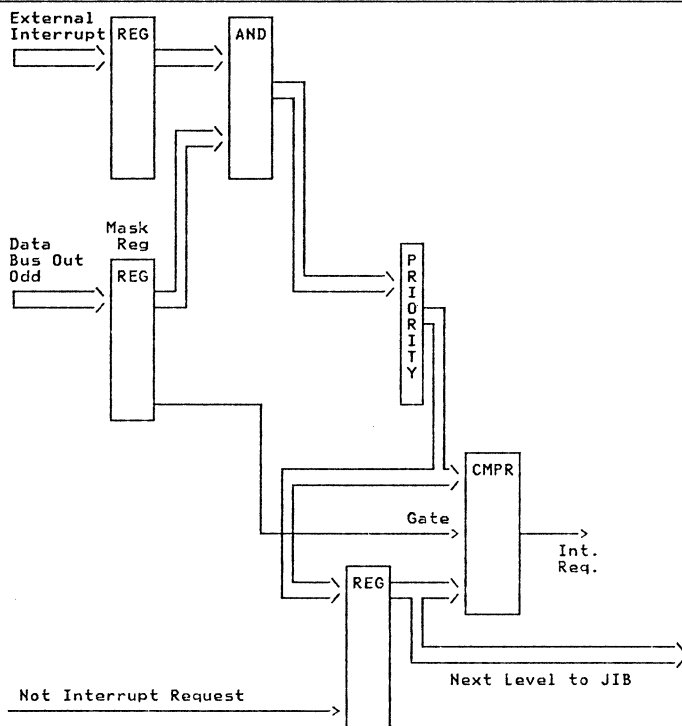


Figure 7. AP Interrupt Module (8691943)

INTERRUPT LEVEL ASSIGNMENTS

Not all interrupt levels are available for use by the attached device. Level 0 is connected to the JIB' ERROR line from JIB, and is used for the program to handle parity and other errors. Level 1 is connected to the hazela cycle steal interface. Levels 2,3, 4, 5, and 6 are available at the AP edge connectors. Level 7 is always active and may be used for background processing.

IBM Internal Use Only

LEVEL	FUNCTION
0	MACHINE CHECK
1	Address Compare (hazela C/S Address Match)
2	Not assigned - wired for attached CSS device
3	Not assigned - wired for attached CSS device
4	Not assigned - available for attached CSS device
5	Not assigned - available for attached device
6	Not assigned - Monitor idle loop
7	Always active - for background processing.
Level 0 is the highest priority level.	

Figure 8. AP Interrupt Level Assignments

INTERRUPT FROM hazela TO THE AP

The address detector within the AP detects that hazela has written to the trigger word location. This causes a level 1 interrupt to the AP, which can then read the contents of the trigger to determine the requested function. This will normally be a pointer to a Data Control Block (DCB), or a similar structure.

The trigger words are located in the block of hazela memory between X'00010' and X'00017'. This allows up to 8 AP's to be attached to one hazela processor. The location assigned to any AP is personalized by the motherboard slot. The low order 3 bits of the trigger address are available to the AP in bits 5-7 of XR X'1B'.

Note: For the hazela to interrupt the AP, the hazela must store to this location with the memory extension bits set to 0. In other words, if the store is done with a CS0 or CS1 instruction, the hazela's I register must be set to 0. If the store is done with a CS1E instruction, the D register must be set to 0. If the memory extension register is not 0, no AP interrupt will occur.

Note: Only the hazela can interrupt an AP in this manner. There is no way for one AP to interrupt another using this mechanism.

Care must be taken in interlocking the AP and hazela programs, since while the AP is executing in level 1, interrupts from hazela to the AP will be lost. For this reason, the AP program should inform hazela that it has left level 1 by manipulating appropriate bits in hazela memory.

hazela INTERRUPTS

Bit 4 of XR X'1C' generates the interrupt from the AP to hazela. This bit is totally under the control of the AP program, so program interlocks must be used to inform the AP program that hazela has accepted the interrupt. A 1 in this bit causes the hazela to be interrupted. POR resets this bit to 0.

Any level of hazela interrupt may be used, but level 4 is suggested.

IBM Internal Use Only

PROGRAMMING CONSIDERATIONS

In order for the AP and hazela to communicate without data loss, it is recommended that the software interface contain interlocks so that:

- hazela is informed that the AP has received a command.
- The AP is informed that hazela has received an interrupt.
- hazela may determine, in a multiple AP system, which AP has generated an interrupt.

If the hazela interrupt is used, it is necessary that software provide a mechanism for the hazela to inform the AP that the interrupt has been recognized, so that the AP can drop the interrupt. For instance, the value X'0000' could be a reserved trigger word value, indicating that the AP interrupt has been recognized.

HAZELA CYCLE STEAL INTERFACE

OVERVIEW

The hazela cycle steal interface is provided to allow attachment of the AP to the hazela engine as a tightly coupled processor.

This interface attaches the AP directly, as a cycle steal device, to the hazela engine. This interface includes an interrupt line to hazela and an address detect circuit to interrupt the AP when hazela writes to a predetermined address in its memory (the Trigger word). The location of the trigger word is determined by the slot on the ARIEL motherboard.

The AP only steals cycles from the hazela to fetch or store data in the hazela memory. The AP cannot execute instructions stored in hazela memory, unless those instructions are first moved from the hazela memory into the AP memory, and then executed from the AP memory. The AP is used as an IPL device for the hazela. A line to reset the hazela engine is provided, so that the AP can prevent hazela from starting during AP bringup and IPL load. A switch is provided, which the AP can sample, to control whether or not the AP is to IPL HAZELA from a series /1 host or from the system memory card EPROMs.

A block diagram of the hazela memory interface is shown in Figure 5 on page 10.

The AP steals cycles from the hazela memory by using an extension to the standard JIB' CS1 instruction. The address in hazela memory is formed in the usual manner for JIB' CS1 instructions, except that the address is extended to 19 bits from the normal 16 by the CS extension bits in XR X'1C'. The CS1 instruction for JIB' has the low order bit (Bit 15) reserved. As JIB' does not use this bit (It may be on or off), hardware external to the JIB' may use this to select whether the AP or the hazela memory is to be used for data.

Two prioritized cycle steal request lines are provided. At T7 time of each JIB' cycle, the clock circuits interrogate the cycle steal lines. If at that time a cycle steal request is active the clock will suspend JIB' timing (TA,TB,TC,TD) and raise the corresponding cycle steal acknowledge. The cycle steal adapter selected uses cycle steal acknowledge as a gate to place address and (if a write) data on the cycle steal address and data busses and activate the read/write line. The clock circuits will generate the timing pulses for memory and will drop cycle steal acknowledge in preparation for the next cycle. The cycle steal adapter must drop cycle steal request before T6 time to prevent taking the next cycle. If the cycle steal adapter is reading data from memory, a strobe is provided for latching the data. See Figure 10 on page 25 for the relationship of the external interface timings to the internal AP clocks.

IBM Internal Use Only

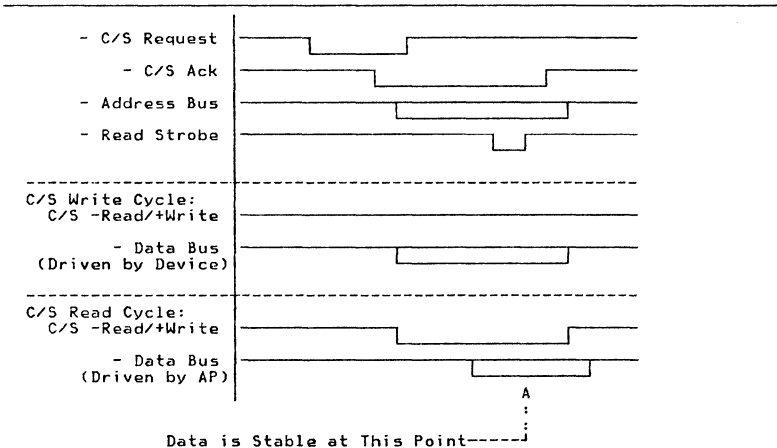


Figure 9. AP Cycle Steal Timing

PROGRAMMING CONSIDERATIONS

EPROM PROGRAMMING

The address lines to memory on the Attached Processor are active low (a low level is equivalent to an address bit value of B'1'), as are the memory data lines. This fact must be noted when writing the Intel 2764 EPROM's which normally will be used to hold the AP program.

INSTRUCTION SET DESCRIPTION

The AP uses the JIB' instruction set, with modifications to the CS1 instruction to permit convenient access to hazela memory. These modifications are described in the following section.

IBM Internal Use Only

CONTROL STORAGE OP 1 (CS1)

Format:

0	1	0	0	1	I	F	0	SAR	0	LSR	0
0					5 6 7 8				1 1	1	
									1 2	5	

Function:

LSR $\leftarrow$ --CS, F=1 or
 CS $\leftarrow$ --LSR, F=0
 SAR $\leftarrow$ --SAR+1, I=1 or
 SAR unchanged, I=0

Where:

I = Increment SAR (Bit 5 = 1)
 F = Memory data fetch (Bit 6=1) or store cycle (Bit 6 = 0)
 SAR = Storage address register even/odd pointer
 LSR = Data register even/odd pointer

Mnemonics:

LD LSR,SAR Load
 LDN LSR,SAR Load with SAR increment
 ST LSR,SAR Store
 STN LSR,SAR Store with SAR increment

This version of the instruction operates exactly as described under the JIB' instruction set.

IBM Internal Use Only

CONTROL STORAGE OP 1 hazela (CS1H)

Format:

0	1	0	0	1	I	F	0	SAR	0	LSR	1
0					5 6 7 8				1 1	1	
									1 2	5	

Function:

LSR $\leftarrow$ --hazela CS, F=1 or
hazela CS $\leftarrow$ --LSR, F=0
SAR $\leftarrow$ --SAR+1, I=1 or
SAR unchanged, I=0

Where:

I = Increment SAR (Bit 5 = 1)
F = hazela data fetch (Bit 6 = 1) or store (Bit 6 = 0)
SAR = Storage address register even/odd pointer
LSR = Data register even/odd pointer

Mnemonics:

LDHZ LSR,SAR	Load
LDHZN LSR,SAR	Load and increment SAR
STHZ LSR,SAR	Store
STHZN LSR,SAR	Store and increment SAR

This instruction operates similarly to the CS1 instruction, except that data is fetched or stored to hazela memory. The hazela memory page to be used is contained in bits 5, 6, and 7 of XR X'1C'. This XR is not modified by this instruction.

MEMORY TIMING

The clocks maintain a separate timing chain for the SUNSET memory. This chain is started whenever the JIB' instruction calls for a SUNSET access. The timing chain lasts 2200 nSec. If the chain has not timed out (i.e. from an earlier SUNSET access, or a SUNSET refresh), the access is delayed to ensure that the SUNSET memory cycle is at least 2200 nSec. The SUNSET is not cycled for other than refresh if it is not being accessed.

In addition, the clock maintains the refresh timer for the SUNSET memory, and starts refresh cycles as required. Refresh cycles have priority over SUNSET access cycles.

IBM Internal Use Only

INSTRUCTION TIMING

The following are the instruction execution times in nanoseconds for the AP:

	Execution time (nanoseconds)			
	Fetch From	Nom	Min	Max
All 1 cycle instructions	RAM	2000	1800	4000
All 1 cycle instructions	R05	550	550	550
CS1H instruction	R05	1300	1050	2550 ¹
CS1H instruction	RAM	2750	2300	3800 ¹
CS1 or CS0 instruction	R05	2550	2350	4550
CS1 or CS0 instruction	RAM	4000	3600	5800
PSW swap	---	1100	1100	1100
Other 2 cycle instructions	R05	1100	1100	1100
Other 2 cycle instructions	RAM	2550	2350	4550
¹ The maximum execution time of this instruction is dependant on what other cycle steal devices are attached to hazela. The given time assumes that the AP has the highest cycle steal priority on the hazela cycle steal.				

IBM Internal Use Only

ERROR CHECKING

GENERAL

On the occurrence of a check condition detected by the microprocessor, the check is latched within JIB'. At the time of occurrence of a check the ERROR output line is raised by JIB'. This line is hard wired on the AP card to cause interrupt level 0 request with a subsequent PSW swap to that level. The error handling routine on level 0 (highest priority level) can then examine the error conditions stored in the PSW area of the LSR's for the level causing the error. The error condition bits are not restored to JIB' when the PSW is returned for the continuation of the program on the error level.

The condition code bit can be examined by looking at the FIRST nibble of the second word of the PSW for all levels of execution. If the AP is in a checked condition the registers are presented on the 3101 console at the time of occurrence, with no user operation required.

These registers can also be interrogated at anytime or for "HUNG" conditions by entering a resident debug facility activated by the 3101 key sequence of ALT and PF1. Refer to the hazel with AP/CSS DEBUG TOOL section of the EDX/J LANGUAGE document for further information.

Note: The PSW swap to level 0 is gated by the state of the interrupt mask bits. The PSW swap will not occur if either the master mask is off or if bit 0 of the interrupt mask is off.

JIB' DETECTED CHECKS

The standard error checks are performed by JIB'.

SUNSET MEMORY SPECIAL CONSIDERATIONS

The programmer should note that the SUNSET memory has a non-zero soft error rate. This means that the memory may generate a parity error, but when the location in question is re-written, the error does not re-occur. This error is caused by alpha particles hitting and discharging the memory cells. If data is sufficiently important that a parity error will cause problems, the programmer should keep two copies of it in the SUNSET memory, and transfer to the other copy if a parity problem is found. Interrupt level 0 may be masked off, and the presence of that interrupt used to determine that an error has occurred.

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

CLOCK

The clock circuitry generates all the timing waveforms required by the JIB' microprocessor and Control Storage. It is driven by a 20 MHz crystal oscillator. Figure 10

JIB' TIMING

TA, TB, TC and TD are eight non-overlapping subcycle time periods required by the JIB' microprocessor. During cycle steal these waveforms are held down effectively stopping the microprocessor for one cycle while the external cycle steal device accesses memory.

The hardware of the clock will ensure that the memory access timings are met by stretching JIB' T1 time. JIB' samples memory data at the end of T2. To fulfill this requirement, when instructions or data are being accessed from EPROM or RAM memory, one machine cycle time is 550 nSec.

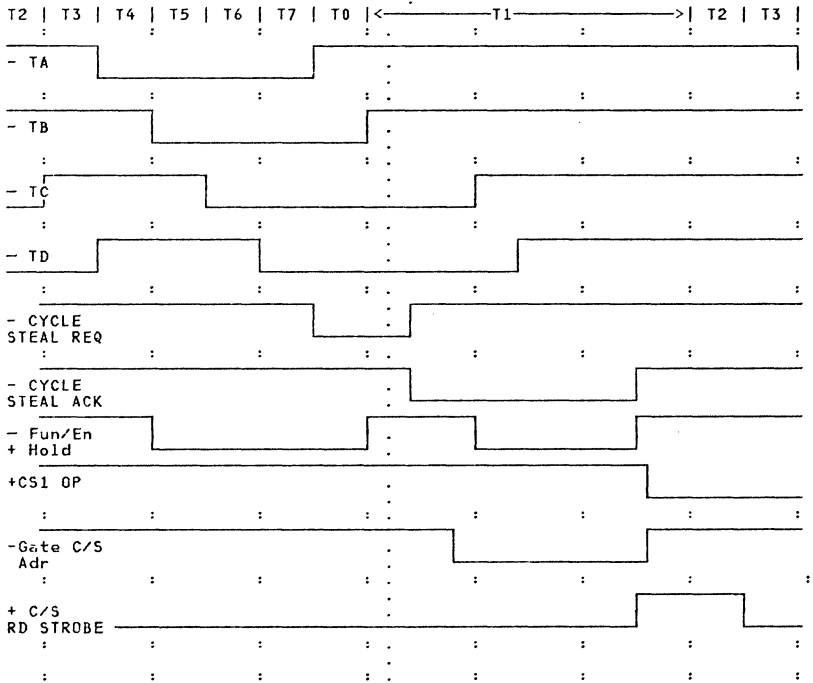


Figure 10. Clock Module Timings: CS1 Op will go active at T2 if CS Select is active.

IBM Internal Use Only

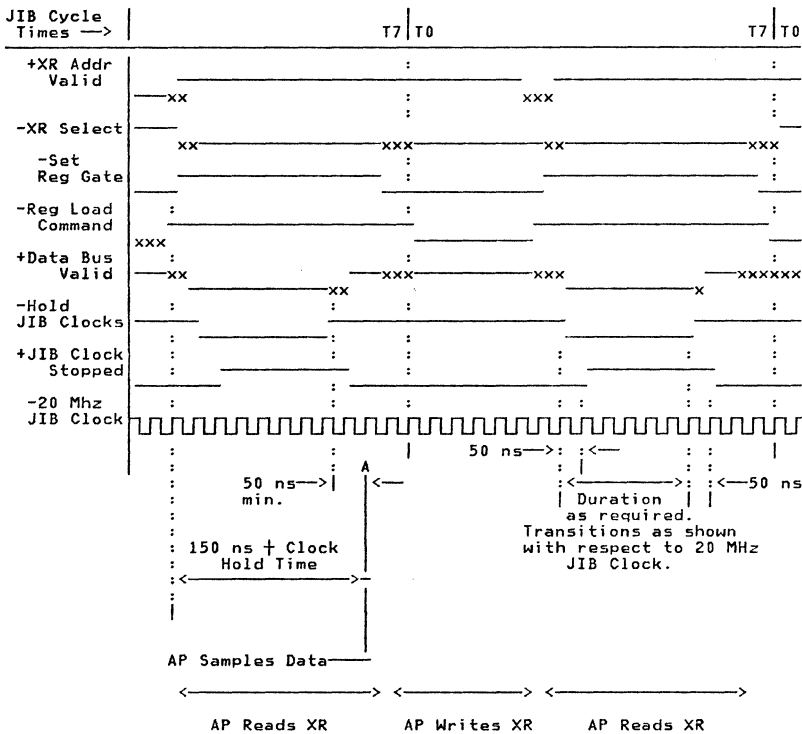


Figure 11. External Register Interface Timing: Note: The AP may cycle the XR's between Read and Write as fast as shown here. It is possible to have "dead" periods between read and write cycles as determined by the AP program.

IBM Internal Use Only

DIAGNOSTICS

The AP test plan takes advantage of the on card resident bringup diagnostics (EPRDM), that are executed every time the system is powered on. The AP/CSS will execute a series of tests in increasing difficulty to completion or to a branch to himself in a hang loop if a test fails.

The AP diagnostic test results are reported via four LEDs located on the top edge of the card. The value is read as a binary number with the low order bit nearest the end plate of the card. The LED display is set to a 0 (all lights off) on power on reset (POR). The value displayed will be changed by the diagnostic programs. These LEDs are controlled by external register '6' (#diag).

AP/CSS LEDs		TEST FAILURE MODE	
HEX BINARY			
F	1111	AP/CSS processor failed to start, no clocks???	
E	1110	AP/CSS processor instruction test failed	
1	0001	AP/CSS LSR to LSR test failed	
2	0010	AP/CSS external register immediate test failed	
3	0011	AP/CSS miscellaneous processor instruction test failed	
4	0100	AP/CSS control store direct test failed	LOAD & STORE instructions.
5	0101	AP/CSS control store indirect test failed	LOAD & STORE instructions.
6	0110	AP/CSS processor instruction test failed	BALR & JUMP instructions.
7	0111	AP/CSS LSR ripple bit test failed	
8	1000	AP/CSS local RAM address or data bit test failed	
9	1001	AP/CSS external register bit test failed	
FLASHING LED COMBINATIONS:			
C	1100	AP/CSS interrupt level switching test failed	
A	1010	AP/CSS cycle steal interface test failed	
B	1011	AP/CSS cycle steal addressing test failed	
C	1101	AP/CSS tests completed	
xx00		CSS-A interface test failed	flashing LEDs on and off.

Figure 12. AP LED DIAGNOSTICS

All blinking LEDs are related to testing the CSS-A card or the clock interface between them.

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

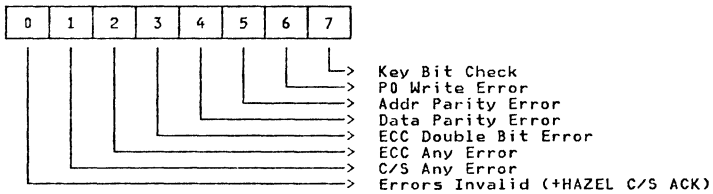
EXTERNAL REGISTER NAMES AND BIT ASSIGNMENTS

External registers shown are for stack 0. Those not shown are not implemented. All external registers for MPCA modules remain the same as HAZEL-A CSS card.

External Register X'04'

Register name - #MSTAT1

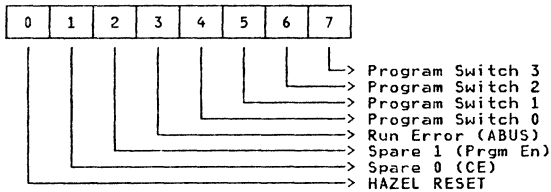
JIB' Access - Read Only



External Register X'05'

Register name - #MSTAT2

JIB' Access - Read Only



User switch 1 determines:

- 0 - EPROM IPL
- 1 - HOST IPL

User switch 2 determines:

- 0 - 4 PORT CONFIGURATION (stacks=0,1,2,4,5)
- 0 - 2 PORT CONFIGURATION (stacks=0,1,2)

User switches 3,4 determines:

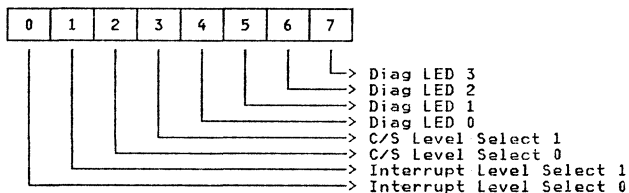
- 00 - BAUD RATE 9,600 BITS/SEC
- 01 - BAUD RATE 19,200 BITS/SEC
- 00 - BAUD RATE 38,400 BITS/SEC
- 00 - BAUD RATE 76,800 BITS/SEC

IBM Internal Use Only

External Register X'06'

Register name - #DIAG

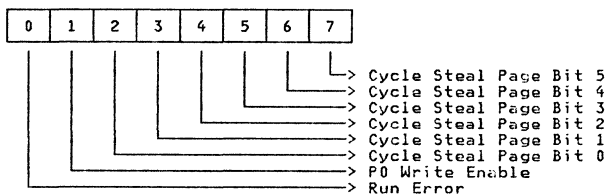
JIB' Access - Read/Write



External Register X'07'

Register name - #HCSADR

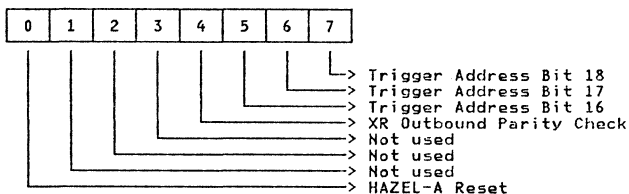
JIB' Access - Read/Write



External Register X'1B'

Register name - #HTRIG

JIB' Access - Read Only

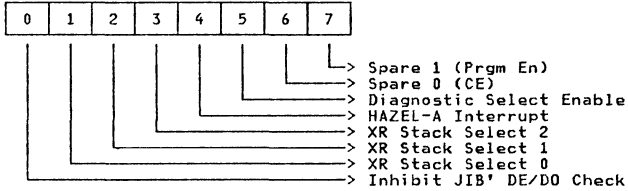


IBM Internal Use Only

External Register X'1C'

Register name - #STKREG, or #HINTREG

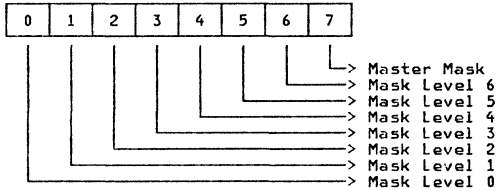
JIB' Access - Read/Write



External Register X'1D'

Register name - #INTM

JIB' Access - Read/Write

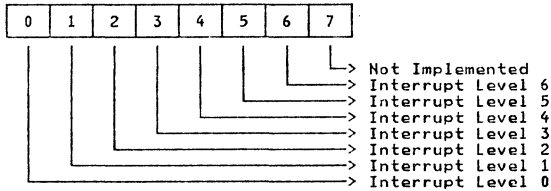


External Register X'1E'

Register name - #INTR

JIB' Access - Read/Write

May be set by external action



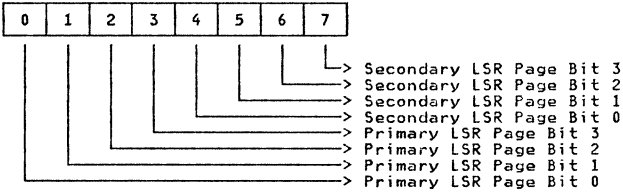
IBM Internal Use Only

External Register X'1F'

Register name - #LSPR

JIB' Access - Read/Write

May be set by PSW swap



Heading ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Heading References</u>
APXRest	ATT	9	EXTERNAL REGISTER ADDRESS SPACE EXTENSION 10

Figure ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Figure References</u>
sysblk	ARCH	3	1: 3
APblock	ATT	7	2: 6
APstruc	ATT	8	3: 6
APXRadr	ATT	9	4: 9
APifHAZ	ATT	10	5: 18
APXRs	ATT	12	6: 9
APintG	ATT	16	7: 14
APclock	CLK	25	10: 18, 25
XRtime	XRS	26	11: 11
Alights	ALIGHTS	27	12:

Imbed Trace

Page vi	RELDOS
Page 2	ARCH
Page 4	APCSS
Page 7	ATT
Page 24	CLK
Page 25	XRS
Page 26	ALIGHTS
Page 28	APCSSXR

DSMDOT528W 'KP' ENDED BY DISALLOWED CONTROL WORD '.sc'.
DSMMOM395I '.DSMFIG' LINE 240: .sc
DSMMOM397I '.DSMFIG' WAS IMBEDDED AT LINE 331 OF 'ATT'
DSMMOM397I 'ATT' WAS IMBEDDED AT LINE 30 OF 'MAIN8'
+++DSMGML003W Unordered List prematurely ended by Heading-2 tag. (Page 1
3)
DSMMOM397I '.DSM#MSG' WAS IMBEDDED AT LINE 150 OF '.DSM#RSET'
DSMMOM397I '.DSM#RSET' WAS IMBEDDED AT LINE 40 OF '.DSMHEAD2'
DSMMOM397I '.DSMHEAD2' WAS IMBEDDED AT LINE 375 OF 'ATT'
DSMMOM397I 'ATT' WAS IMBEDDED AT LINE 30 OF 'MAIN8'
DSMBEG323I STARTING SECOND PASS.
DSMDOT528W 'KP' ENDED BY DISALLOWED CONTROL WORD '.sc'.
DSMMOM395I '.DSMFIG' LINE 240: .sc
DSMMOM397I '.DSMFIG' WAS IMBEDDED AT LINE 331 OF 'ATT'
DSMMOM397I 'ATT' WAS IMBEDDED AT LINE 30 OF 'MAIN8'
+++DSMGML003W Unordered List prematurely ended by Heading-2 tag. (Page 1
3)
DSMMOM397I '.DSM#MSG' WAS IMBEDDED AT LINE 150 OF '.DSM#RSET'
DSMMOM397I '.DSM#RSET' WAS IMBEDDED AT LINE 40 OF '.DSMHEAD2'
DSMMOM397I '.DSMHEAD2' WAS IMBEDDED AT LINE 375 OF 'ATT'
DSMMOM397I 'ATT' WAS IMBEDDED AT LINE 30 OF 'MAIN8'

ARIEL USER MANUAL (I/O CARDS)

Control Systems Engineering

**IBM Corporation
General Technology Division
D-035 B-300-47A
East Fishkill
tel 533-7654**

IBM Internal Use Only

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

TABLE OF CONTENTS

Ariel User Manual	1
Communications	1
Transaction Format	1
Communication Data	2
Destination Communications Card (DCA-A)	2
Switch	3
32 Bit DI/DO-A	4
Input Jumpers	4
Output Jumpers	5
A/D-A	6
Jumper Banks A, B, and C	6
Resistor Bridge A/D-A	7
Calibration	7
DAC-A	9
Jumpers	9
Calibration	10
TPAC-A	11
8 Position Dip Switch	11
10 Position Tap Switch	12
Single Pole Double Throw Switches	13
Jumper Selection	14
EPROM4-A	15
Jumpers	15
24 Volt Power Connector	16
Appendix A. Connector Information (All Cards)	17
CTC Definitions (All cards)	19
Software Available For TPAC-A	27
Test Cables (All Cards)	27

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

LIST OF ILLUSTRATIONS

Figure 1. Selector Switch (8 position dip switch)	3
Figure 2. Digital Input Circuit	4
Figure 3. Digital Output Circuit	5
Figure 4. A/D-A Jumper Configurations and Definitions.	6
Figure 5. Resistor Bridge and Channel Layout	7
Figure 6. DAC-A Jumper Position Definitions	9
Figure 7. Jumper Configuration	15
Figure 8. Power Connector Placement	16
Figure 9. User Pin Definitions and Assignments	18
Figure 10. User Pin Definitions and Assignments	19
Figure 11. A/D-A Test Cable	28
Figure 12. DAC-A Test Cable	29
Figure 13. 32 Bit DI/DO-A Test Cable	29
Figure 14. TPAC-A Test Cable For POM CODE	30

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

ARIEL USER MANUAL

This document will describe and collect all information, about I/O cards for Ariel, pertinent to the user and organize it in a suitable manner for ease of access.

Certain information concerning communications to the I/O cards and also technical data about each card will be discussed, but only in a general manner. For more details on a specific card see the Functional Specification of that card.

The cards that will be covered are:

Desc	Asm P/N	RC P/N	Sch P/N
DCA-A	6912784	6912785	6912786
TPAC-A	6912778	6912779	6912780
DI/D032-A	6912757	6912758	6912759
A/D-A	6912766	6912767	6912768
DAC-A	6912769	6912770	6912771
EPR0M4-A	71X8902	71X8903	71X8904

With reference to the Mother Board for Ariel, the AP/SCA (Attached Processor - Source Communications Adapter), and the DCA (Destination Communications Adapter).

COMMUNICATIONS

Communication to the I/O cards from the host is implemented through a 93 ohm COAX cable. The host transmits data and it is received at the RDIS-A via the DCA. The DCA then retransmits the data to the next RDIS-A and, if the data is for him, will decode the information and present it to the proper I/O card. This is detailed in the DCA-A Functional Specification. Host communications is detailed in the AP/SCA Functional Specification.

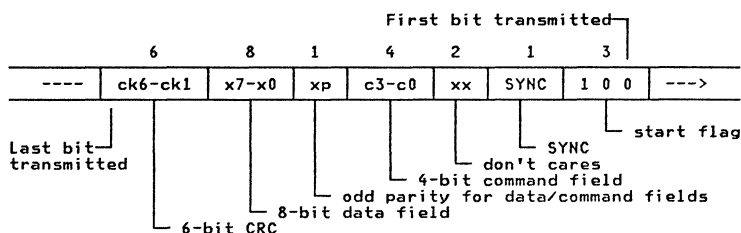
Note: If power is lost in a RDIS-A, relays on the DCA will allow communication to continue to all other functional blocks.

Transaction Format

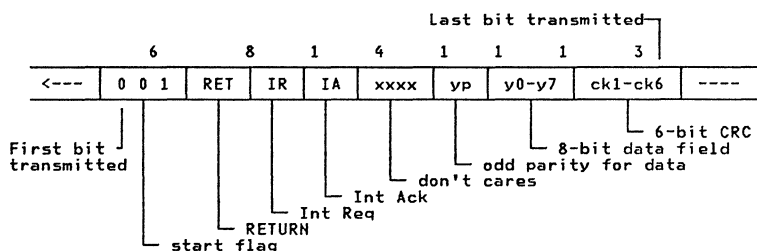
All transactions between the host processor and RDIS-A blocks are sent in small packs of 25 bits in length. Each pack includes a 3 bit start flag, a 7 bit control field, an 8 bit data field w/parity, and a 6 bit CRC code for the data and control fields.

The format for a message from the host processor to a RDIS-A block is given below.

IBM Internal Use Only



The format for the return message from the RDIS-A blocks back to host processor is given below.



Communication Data

Maximum Distance 5000 feet of RG-62 Coaxial Cable
 Data Rate 1.25 Mbaud (Million bits/sec)
 Data Format 25 Bit Asynchronous
 Data Code Biphasic Mark (Self-Clocking)
 Transmit Amplitude 8 volts peak-peak max
 Receiver Sensitivity 0.7 volts peak-peak min
 Repeater Delay 400 nsec

DESTINATION COMMUNICATIONS CARD (DCA-A)

Notes -

1. During Power off, relays allow communications to downstream units
2. 20 position connector for I/O Interrupts and LSA Selects
3. 120 position connector for Local Data Bus and Power
4. Utilizes the UBI module
5. Coax connector at bottom of card is input from host or upstream unit
6. Coax connector at top of card is output to downstream units
7. Coax cable -> is RG-62 AU 93 ohm

IBM Internal Use Only

Switch

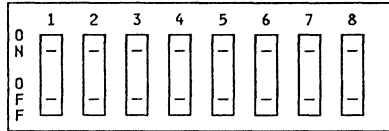


Figure 1. Selector Switch (8 position dip switch)

Switch 1-5 Block Address, Switch 1 -> MSB, Switch 5 -> LSB

On is active (gnd)

Switch 6 Mode of operation of the UBI Module

- ON - Parallel Mode
- OFF - Serial Mode

Note: Switch 6 should always be off

Switch 7-8 Sets the timing interval on the Operational Monitor (OP-MON)

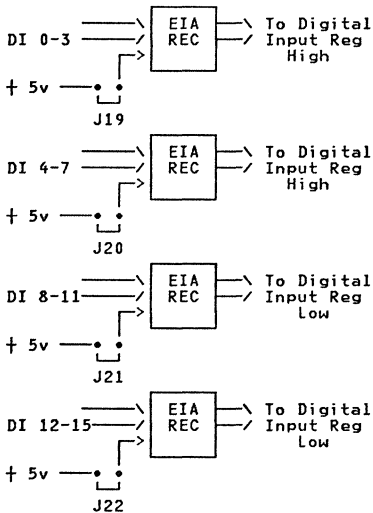
Switch 7	Switch 8	
OFF	OFF	-> .105 Sec time out
OFF	ON	-> .420 Sec time out
ON	OFF	-> 1.68 Sec time out
ON	ON	-> 3.35 Sec time out

IBM Internal Use Only

32 BIT DI/DO-A

Input Jumpers

The digital input circuit provides capability to receive standard RS232C EIA inputs or VTL inputs. The inputs are gated through 4 EIA receivers each with their own program jumper. If the jumper is on, the receiver can accept only EIA signals. If the jumper is off, the receiver can accept only VTL inputs. See Figure 2.



J	Jump Pos	Function	DI Bits
J19	on	EIA Rec	0-3
J20	on	EIA Rec	4-7
J21	on	EIA Rec	8-11
J22	on	EIA Rec	12-15
J19	off	VTL Rec	0-3
J20	off	VTL Rec	4-7
J21	off	VTL Rec	8-11
J22	off	VTL Rec	12-15

VTL Mode Threshold Voltages
low → + 1.4 volts
high → + 2.2 volts

EIA Mode Threshold Voltages
low → - 1.2 volts
high → + 2.2 volts

Figure 2. Digital Input Circuit: If the jumper is off the receiver can accept both EIA, and VTL. But if in VTL mode and input is EIA, the data being read may be invalid.

IBM Internal Use Only

Output Jumpers

The digital outputs are driven through 64 ma drivers with jumpers for each. If the jumper is off, the load resistor (pull-up) on the digital output can sink 64 ma. If the jumper is on, the max load resistance cannot allow more than 29 ma ($(64\text{ma} - (5/200)\text{ma} - 10\text{ma})$). The 10 ma is a safety margin for the DI/D032-A card.

Not more than EIGHT digital outputs can sink 64 ma at any one time. But all 16 DOs can sink 30 ma at any given time. See Figure 3

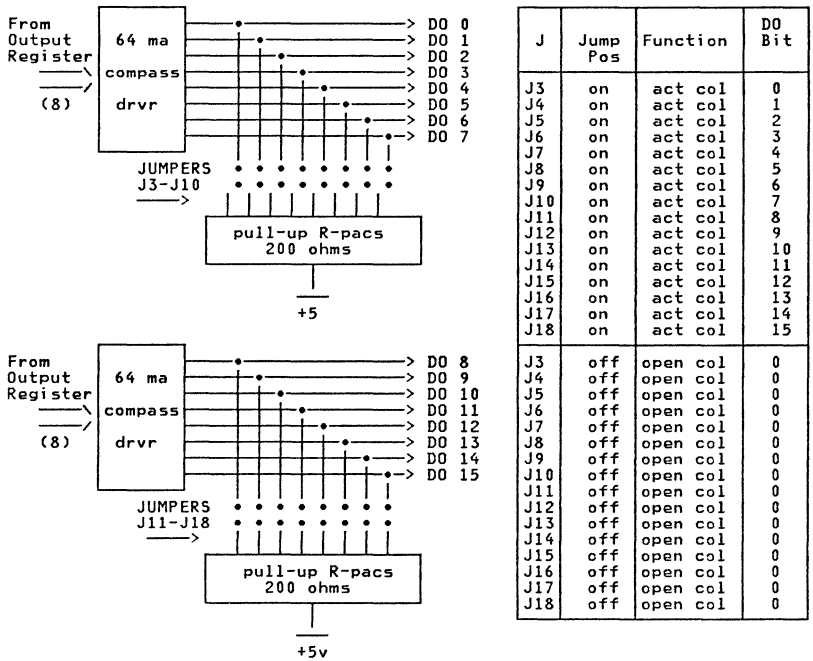


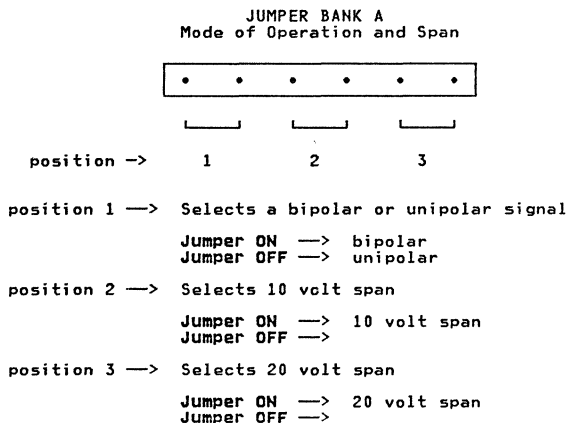
Figure 3. Digital Output Circuit

IBM Internal Use Only

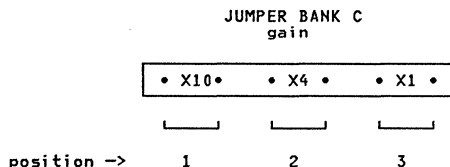
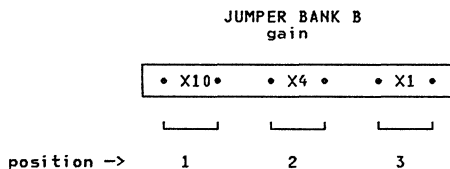
A/D-A

Jumper Banks A, B, and C

There are 3 sets of jumper banks on this card. They are used for selecting the mode of operation and the gain. These jumper banks are located near the end of the card farthest from the 37 pin D-Shell.



Jumpers cannot be on position 2 and 3 simultaneously
A 20 volt span unipolar signal cannot be used



If no gain is desired the jumpers must be on the X1 position
The input voltage x gain cannot exceed 10 volts

Figure 4. A/D-A Jumper Configurations and Definitions.

IBM Internal Use Only

The hardware programmable gain is selected via placing jumpers on the appropriate jumper banks as shown. There may only be one jumper on each bank (B and C) and the gains selected are multiplicative.

Resistor Bridge A/D-A

The A/D-A card also provides a socket and a resistor bridge plugged into that socket, located near the 37 pin D-Shell. The A/D-A can accept 4-20 ma signals through the analog channels that have resistors soldered in the socket corresponding to that channel.

It is left up to the user to configure resistors on the resistor bridge and also to document the configuration.

Note: If a resistor is soldered across a channel on the resistor bridge, then that channel can accept only a current input signal.

Note: If there is no resistor across a channel then that channel can accept only a voltage signal.

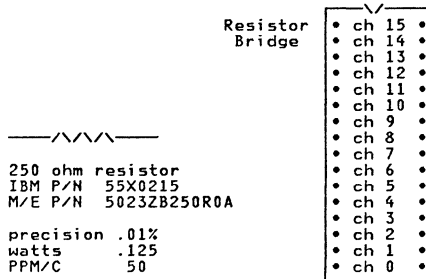


Figure 5. Resistor Bridge and Channel Layout: The 250 ohm resistor can be ordered from "HEPCO/ELECTRA, INC. AIRPORT RD. PO BOX 760 MINERAL WELLS, TEXAS 76067"

Calibration

To Calibrate the following is needed.

1. Digital Volt Meter

The more accurate the DVM, the more accurate the calibration will be. A DVM accurate to 10 mv is the minimum requirement.

2. 37 Pin D-Shell Male Jumper Cable wired as follows.

Pin 19 to Pin 18
Pin 37 to Pin 17 and Pin 14
Pin 22 to Pin 16
Pin 23 to Pin 15

3. Small Screw Driver to adjusting POTs

4. A/D-A Card P/N 6912766

Configured for +/- 10 volt range and hardware Gain = 1. See Figure 4 on page 6

IBM Internal Use Only

Using the RDIS-A Test Software (GCODE) the user must select the AI option (Analog Input Test). Once the Main menu is on the screen, select option 3 (Change LSA) to address the A/D-A card (option 1). Then return to the Main menu and select

UTILITIES	(option 4), then select
ADUTL	(option 5),

Now begin Calibration;

• Step 1 - Zero Adjust

- | Description | Menu Option |
|--|------------------------|
| 1. Select Loop 1 Channel | (1) |
| 2. Select Channel = 0 | (0) |
| 3. Select Gain = 1 | (1) |
| Adjust R5 until TP8 = 0.0000 volts. Use a Digital Volt Meter across TP8 (+) and TP10 (Agnd). | |
| + 0.0049v = hex '001' | --- For reference only |
| + 0.0000v = hex '000' | |
| - 0.0049v = hex 'FFF' | |
| Adjust R6 until TP9 = 0.0000 volts. Use a Digital Volt Meter across TP9 (+) and TP10 (Agnd). | |
| + 0.0049v = hex '001' | --- For reference only |
| + 0.0000v = hex '000' | |
| - 0.0049v = hex 'FFF' | |
| 4. Attention EL | (Alt PF7, EL) |

• Step 2 - Gain Adjust Positive

- | Description | Menu Option |
|---|---------------|
| 1. Select Loop 1 Channel | (1) |
| 2. Select Channel = 1 | (1) |
| 3. Select Gain = 1 | (1) |
| Adjust R10 until 3101 Screen Read 9.9998 volts. hex '7FF' | |
| 4. Attention EL | (Alt PF7, EL) |

• Step 3 - Gain Adjust Negative

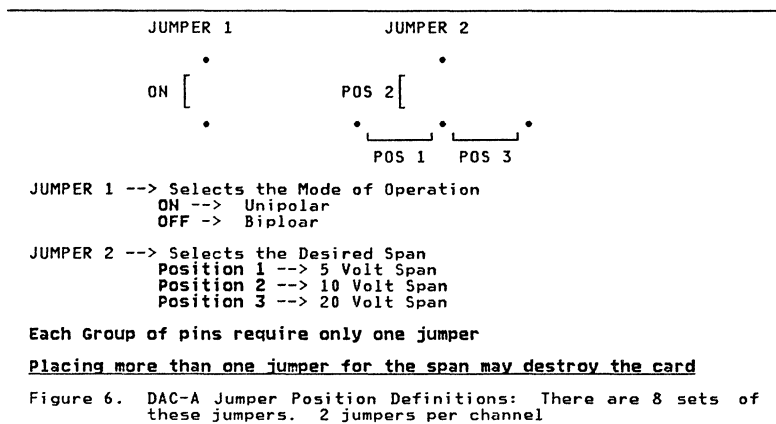
- | Description | Menu Option |
|---|---------------|
| 1. Select Loop 1 Channel | (1) |
| 2. Select Channel = 2 | (2) |
| 3. Select Gain = 1 | (1) |
| Adjust R8 until 3101 Screen Reads - 9.9998 volts. hex '800' | |
| 4. Attention EL | (Alt PF7, EL) |
| The Card is now Calibrated for all channels. | |

IBM Internal Use Only

DAC-A

Jumpers

The DAC-A has two jumpers for each channel. One selects the channel as bipolar or unipolar, and the other selects the range as 5v, 10v, or 20v. These jumpers are located near the top edge of the card. Channel 0 jumpers are furthest from the 25 pin D-Shell and channel 8 is closest to the 25 pin D-Shell.



IBM Internal Use Only

Calibration

Using the RDIS-A Test Software (GCODE) the user must select the A0 option (Analog Output Test). Once the Main menu is on the screen, select option 3 (Change LSA) to address the DAC and the A/D-A. Return to the Main menu and select

DAC Utilities (option 4), then select
DAC to A/D Wrap (option 4),

Now begin Calibration;

• Step 1 - Zero Adjust

Description	Menu Option
1. Select Channel 0	(0)
2. Select Range of 0 - 5 v	(1)
3. Select Continuous Loop	(-1)
4. Select Repeat	(1)
5. Select 0v as output voltage	(0)

Monitor channel 0 output with a DVM and adjust the potentiometer marked Z (ZERO) for channel 0 until the meter reads 0.00 volts.

6. Attention EL (alt PF7 EL)

• Step 2 - Gain Adjust

Description	Menu Option
1. Select Channel 0	(0)
2. Select Range of 0 - 5 v	(1)
3. Select Continuous Loop	(-1)
4. Select Repeat	(1)
5. Select 5v as output voltage	<u>(5.0)</u>

Monitor channel 0 output with a DVM and adjust the potentiometer marked G (GAIN) for channel 0 until the meter reads 5.00 volts.

6. Attention EL (alt PF7 EL)

Repeat from step 1 for the remainder of the eight channels.

IBM Internal Use Only

TPAC-A

There are three different types of switches on this card. The first is a simple 8 position dip switch. The second is a 10 position TAP switch, and the third is a single pole double throw (spdt) switch. The third switch multiplexes two inputs to a common signal. Also on the card is a jumper for communications.

8 Position Dip Switch

This switch controls miscellaneous hardware functions on the card. It should be noted that once set up they need not be altered unless for debug purposes.

Switch 1 Selects high/low byte of Timer module counter to Xreg9. See the section on the tap switch for selection of actual counter frequency.

1. **ON** - Enables 8 - 15 count to mux (high frequency)
2. **OFF** - Enables 0 - 7 count to mux (slow frequency)

Switch 2, 3 Enables modem operation of RS232 port

1. **2 ON and 3 OFF** - Request to Send (RTS) tied to Clear to Send (CTS)
2. **2 OFF and 3 ON** - Request to Send separate from Clear to Send

Switch 4 Not Used

Switch 5 Not Used

Switch 6 Controls EIA receiver threshold of Receive Data

1. **ON** - high - input circuit receives EIA signals
2. **OFF** - open - input circuit receives VTL signals

Switch 7 Controls EIA receiver threshold of EIA Clear to Send (CTS)

1. **ON** - high - CTS input must be an EIA signal
2. **OFF** - open - CTS input must be a VTL signal

Switch 8 Controls EIA receiver threshold of EIA Data Set Ready (DSR)

1. **ON** - high - DSR input must be an EIA signal
2. **OFF** - open - DSR input must be a VTL signal

IBM Internal Use Only

10 Position Tap Switch

This Switch controls which clock output of the Timer module will be available to the multiplexer of Xreg9.

Switch	Dip Switch 1 OFF	Dip Switch 1 ON
Position 1	2.543 Hz	651.04167 Hz
Position 2	5.086 Hz	1302.0833 Hz
Position 3	10.1725 Hz	2604.1667 Hz
Position 4	20.3450 Hz	5208.3333 Hz
Position 5	40.6901 Hz	10416.667 Hz
Position 6	81.3802 Hz	20833.333 Hz
Position 7	162.760 Hz	41666.667 Hz
Position 8	325.521 Hz	83333.333 Hz
Position 9	+ 5 volts to Xreg9	
Position 10	+ Gnd to Xreg9	

Note: All counter outputs from the Timer module are 50 % duty cycle

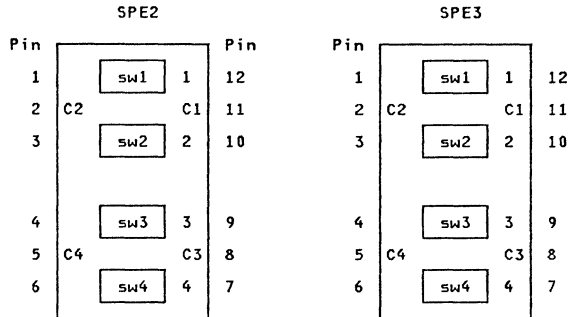
IBM Internal Use Only

Single Pole Double Throw Switches

These switches multiplex the on card counter frequency and DI bits 8 - 15 to Xreg9.

Switch to the right —> On card counter frequency to Xreg9

Switch to the left —> Digital Input to Xreg9



Switch	switch to right	switch to left	
U28			
sw1	counter freq	DI bit 8	-> to Xreg9 bit 0
sw3	counter freq	DI bit 9	-> to Xreg9 bit 1
sw4	counter freq	DI bit 10	-> to Xreg9 bit 2
sw2	counter freq	DI bit 11	-> to Xreg9 bit 3
U29			
sw1	counter freq	DI bit 12	-> to Xreg9 bit 4
sw3	counter freq	DI bit 13	-> to Xreg9 bit 5
sw4	counter freq	DI bit 14	-> to Xreg9 bit 6
sw2	counter freq	DI bit 15	-> to Xreg9 bit 7

The side of the switch which is down defines left or right in the table above

IBM Internal Use Only

Jumper Selection

This jumper selects the type of transmit data for the RS232 port. The Transmit data can be driven by an Isolated Current Loop or EIA.

- ZM1
- POS 1 [
- ZM2
- POS 2 [
- ZM3

Position 1 -> Selects Transmit Data for Isolated Current Loop
Position 2 -> Selects Transmit Data for EIA Standard

Note: These pins are located directly behind the 9 pin D-Shell

IBM Internal Use Only

EPROM4-A

Jumpers

EPROM4-A will be used only as an Eprom Burn card.

There are three sets of hardware jumpers described below.
See Figure 7 for position information.

Group 1 These 2 jumpers are used to select the 256K density eproms

Position 1 --> burn 2764A, 27128A
Position 2 --> burn 27256

Group 2 These 2 jumpers multiplex Chip Enable and Eprom Program

Note: This jumper should be left in Position 1 while executing the Eprom Burn Program (BURN2)

Position 1 --> Chip Enable to Pin 20 of eprom sockets
Position 2 --> Eprom Program to Pin 20 of eprom sockets

Group 3 This jumper selects Eprom group 4 as burn or not to burn

ON --> Eprom Group 4 is programmable
OFF --> Eprom Group 4 is read only

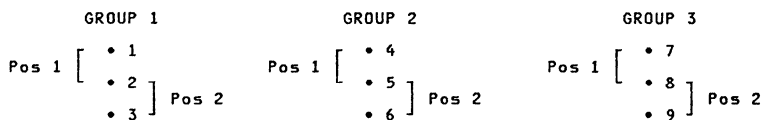


Figure 7. Jumper Configuration: Group 1 and 2 are located near the top left of the card, by the 24 volt cable connector. Group 3 is located between Eprom Group 3 and Eprom Group 4.

IBM Internal Use Only

24 Volt Power Connector

The 24 volts to Eprom4-A is brought in through SPE1 and SPE2. These are right angle BERG headers located on top of Eprom4-A near the back. Both of these headers are wired the same and are inter-changeable. They are wired as shown in Figure 8;

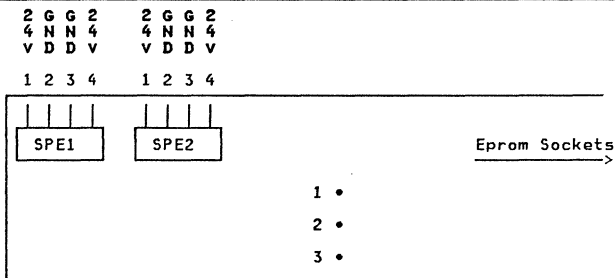


Figure 8. Power Connector Placement: There are two power connectors. If burning multiple cards, the 24 volt power can be jumpered from card to card.

IBM Internal Use Only

APPENDIX A. CONNECTOR INFORMATION (ALL CARDS)

IBM Internal Use Only

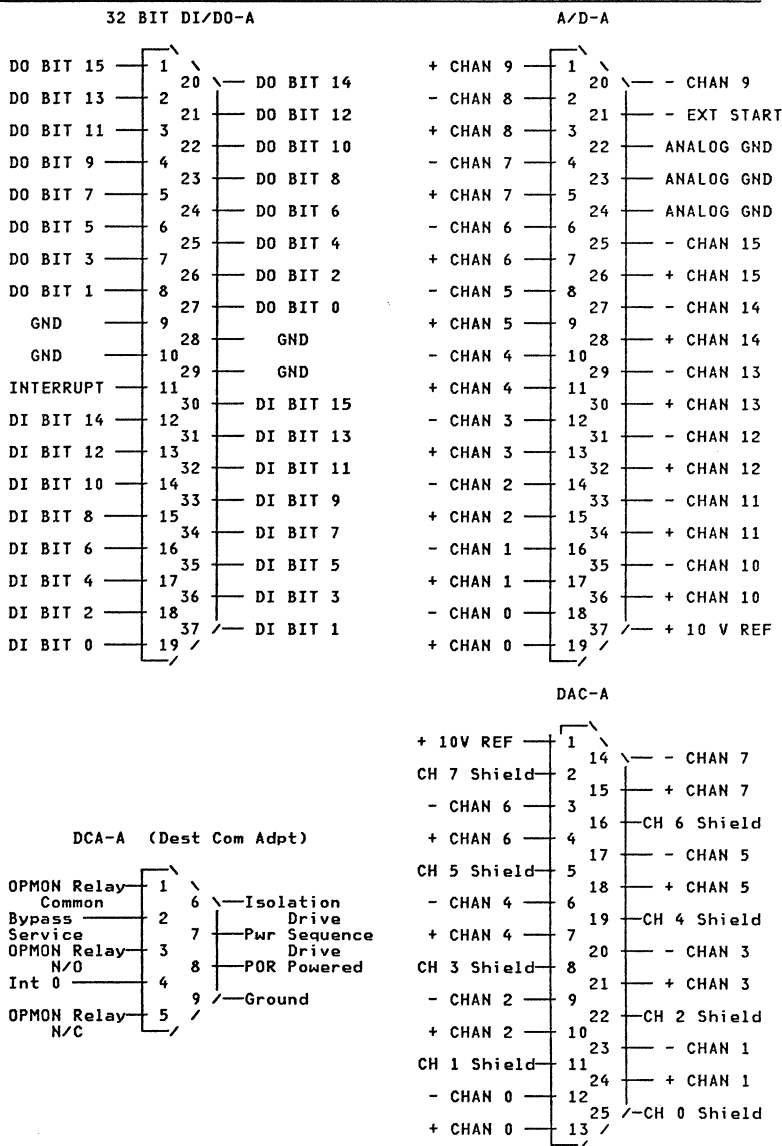


Figure 9. User Pin Definitions and Assignments

IBM Internal Use Only

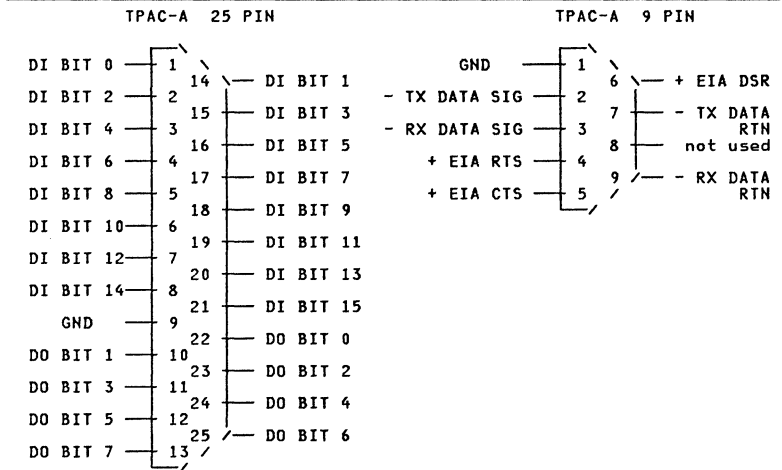


Figure 10. User Pin Definitions and Assignments

CTC DEFINITIONS (ALL CARDS)

• 32 Bit DI/DO-A

See 32 Bit DI/DO-A Functional Specification for more information.

CTC 0	-->	I/O Select	
CTC 4	-->	Read DI 0-7	
CTC 5	-->	Read DI 8-15	
CTC 6	-->	Read DO 0-7	
CTC 7	-->	Read DO 8-15	
CTC A	-->	Write DO 0-7	
CTC B	-->	Write DO 8-15	
CTC F	-->	Read Card ID	(hex '30')
CTC 1	-->	not used	
CTC 2	-->	not used	
CTC 3	-->	not used	
CTC 8	-->	not used	
CTC 9	-->	not used	
CTC C	-->	not used	
CTC D	-->	not used	
CTC E	-->	not used	

• A/D-A

See A/D-A Functional Specification for more information.

CTC 0	-->	I/O Select	
CTC 1	-->	Reset Int Req, Int Pend, Diag Latch	
CTC 2	-->	Read Data Reg High	
CTC 3	-->	Read Data Reg Low	
CTC 4	-->	Write Control Reg High	
		bits 0-3 -->	Analog Channel Select

IBM Internal Use Only

bits 4-5 -> Gain Select
bits 6-7 -> not used
CTC 5 --> Write Control Reg Low
bit 8 -> Start Convert
bit 9 -> Enable Interrupt
bit A -> Allow Ext Start
CTC 6 --> Read Control Reg High
CTC 7 --> Read Control Reg Low

- If Diagnostic Latch Off, (See CTC 9)

Read Status Reg Low
bit 8 -> Interrupt Pending
bit 9 -> End Of Convert (EOC)
bit A -> Interrupt enabled
bit B -> External Start Allowed

- If Diagnostic Latch On, (See CTC 9)

Read Data From Last CTC 5

CTC 9 --> Set Diagnostic Latch
CTC D --> Reset Data Control
CTC F --> Read Card ID (hex '2C')

CTC 8 --> not used
CTC A --> not used
CTC B --> not used
CTC C --> not used
CTC E --> not used

IBM Internal Use Only

- DAC-A

See DAC-A Functional Specification for more information.

CTC 0 --> I/O Select
CTC 1 --> Reset All Outputs and Registers
CTC 2 --> Write Reg High (8 bits of data)
CTC 3 --> Write Reg Low (4 bits of data, 4 bits control)

- Control bits

bit C --> Diagnostic Bit
ON (1) --> Hold Data
OFF (0) --> Write DAC
bits D-F --> DAC Channel Selected

CTC 6 --> Read Reg High
CTC 7 --> Read Reg Low
CTC F --> Read Card ID (hex '1E')

CTC 4 --> not used
CTC 5 --> not used
CTC 8 --> not used
CTC 9 --> not used
CTC A --> not used
CTC B --> not used
CTC C --> not used
CTC D --> not used
CTC E --> not used

IBM Internal Use Only

• TPAC-A POM

See Pulsed Output Modulation Macro-Card Functional Specification for more information.

CTC 0 --> I/O Select

CTC 1 --> Reset Interrupt and Status

CTC 2 --> Write Control Reg High (enable/disable)
bits 0-7 -> SCR0 - SCR7

CTC 3 --> Write Control Reg Low
bits 8-15 -> SCR8 - SCR15

CTC 4 --> Read Status Reg High
bit 0 -> Interrupt Pending
bits 1-3 -> not used
bits 4-7 -> line adr of status being read

CTC 5 --> Read Status Reg Low
bit 8 -> Count Error
bit 9 -> SCR Shorted
bit A -> SCR Open
bit B -> Clock Shorted
bit C -> Clock Open
bit D-F -> Spare

CTC 6 --> Write Duty Cycle Reg (Line Adr)
bits 0-3 -> not used
bits 4-7 -> line adr for duty cycle to be set

CTC 7 --> Write Duty Cycle Reg (Value)

CTC 8 --> Read Duty Cycle Reg (Line Adr)
bits 0-3 -> not used
bits 4-7 -> line adr for duty cycle to be read

CTC 9 --> Read Duty Cycle Reg (Value)

CTC A --> Read Line Status Word (Line Adr)
bits 0-3 -> not used
bits 4-7 -> line adr of status being read

CTC B --> Read Line Status Word (value)
bit 8 -> Count Error
bit 9 -> SCR Shorted
bit A -> SCR Open
bit B -> Clock Shorted
bit C -> Clock Open
bit D-F -> Spare

CTC C --> Read Control Reg (enable/disable)
bits 0-7 -> SCR0 - SCR7

CTC D --> Read Card ID
bits 8-15 -> SCR8 - SCR15

CTC F --> Read Card ID (hex '33')

CTC E --> not used

IBM Internal Use Only

• TPAC-A Auto-Poller

See Auto-Poller Functional Specification for more information.

CTC 0 --> I/O Select

CTC 1 --> Reset Interrupt Req and Status Reg

Resets Status Reg bits 0, 1, 4, 5, 6, F

CTC 3 --> Write ROC

EBCDIC TO ASCII conversion can be done automatically.
This command will be ignored if the status reg indicates
transmit or receive mode, or interrupt is pending.
The data format is as follows;

| Byte Count High| Byte Count Low| Addr| Cmd| Dis Ptr| Data.. Data|

Byte Count Counts in hex the total number of bytes in the transaction (64 data bytes plus header max)

Address Address of ROC Terminal (00-FE)

Address FF is a broadcast to all terminals that receive messages but do not respond

Command Writing ROC

- 10 - Writes display with no mode change
- 20 - Reads display
- 40 - Unlock Terminal
- 60 - Writes Alarm
- 70 - Writes display and puts ROC into Entry
Data Mode with lower panel shifted
- 80 - Writes display and puts ROC into
Function Key Mode
- 90 - Writes display and puts ROC into Entry
Data Mode with no lower panel shifted
- A0 - Write Backlight On
- B0 - Write Backlight Off

Display Pointer Specifies the starting character coordinate on the display

- Hi nibble = one of 4 lines (0-3)
- Low nibble = one of 16 positions (0-F) within the line, where 0 is furthest to the left

- 1) While broadcasting allow a 5msec delay after transmitting data.
- 2) Dis Ptr is not needed for commands 20,40,60.
- 3) Data = 00 for Alarm off, and Data = 01 for Alarm on.
- 4) For commands A0 and B0 the data is the coordinate of the backlight being written to.

IBM Internal Use Only

CTC 5 --> Read Receive Buffer

Used after status reg has been interrogated. Command is ignored if the status reg indicates transmit mode or interrupt pending. Also used to read the address of a terminal in trouble as indicated by the status reg. The data format is as follows;

| Byte Count High| Byte Count Low| Addr| Cmd| Dis Ptr| Data .. Data|

Byte Count Total number of bytes in transaction including byte counts

Address Same as CTC 3

Command Read ROC

- 52 - Data present is from the ROC
- 53 - Data present is from the Magnetic Hand Scanner or the Magnetic Slot Reader. Up to 128 bytes are formatted in EBCDIC. Special characters can be used.
- 54 - Data present is a switch coordinate from the ROC. (hex mode)
- 62 - Data present is the contents of the ROC display. (EBCDIC or ASCII and 1-64 data bytes)

CTC 6 --> Read Status Reg High

- bit 0 -> Interrupt pending
- bit 1 -> Receive mode
- bit 2 -> Transmit mode
- bit 3 -> Excessive NAK's from Terminal
- bit 4 -> Parity Error
- bit 5 -> Framing Error
- bit 6 -> LRC Error
- bit 7 -> Excessive Time Out from Terminal

CTC 7 --> Read Status Reg Low

- bit 8 -> Teleprocessor Line Hung
- bit 9 -> not used
- bit A -> not used
- bit B -> ASCII Table not defined
- bit C -> EBCDIC Table not defined
- bit D -> Poll List not defined
- bit E -> Gas Panel Character Conversion not defined
- bit F -> not used

CTC 8 --> Define the control commands for writing sub-commands to TPAC-A (through the DIS BUS)

- | Sub-Command | Description |
|-------------|---|
| hex '00' | -> Poll Terminals |
| hex '10' | -> Suspend Polling |
| hex '20' | -> Defines ASCII table |
| hex '30' | -> Defines EBCDIC table |
| hex '40' | -> Defines Poll List |
| hex '60' | -> Enables Gas Panel Character Conversion |
| hex '70' | -> Resets Receive Mode bit |

1) Command 70 will only function while Interrupt Pending.

IBM Internal Use Only

CTC D --> Defines the ASCII or EBCDIC tables.

Must be issued after CTC B has been issued with sub commands 20, 30, and 40. The accompanying data with this CTC will be the table data, either ASCII, EBCDIC, or the Poll list. The end of the data string should be marked with an X'FF' delimiter. The appropriate Status Reg bit will be turned OFF at the end of the string. The EBCDIC or ASCII generation must be fully completed for their appropriate status bits to be turned off.

CTC F --> Read Card ID (hex '27')

CTC 2 --> not used
CTC 4 --> not used
CTC 8 --> not used
CTC 9 --> No Application (for Gas Panel only)
CTC A --> not used
CTC C --> not used
CTC E --> not used

IBM Internal Use Only

• TPAC-A SECS

See Vendor Micro-Processor Interface User Guide for more information.

CTC 0 --> I/O Select

CTC 1 --> Reset Interrupt Request and Status Reg

Status reg will be reset to hex '0001'

CTC 3 --> Write Transmit Buffer

The data format is as follows;

Count	Data	Data		Data
-------	------	------	-------	------

Count Number of data bytes in transmission excluding the Count Byte

Data Maximum number of bytes to transmit are 254

Data can be in Binary, EBCDIC, ASCII, or any other format.
The application Program will determine this.

CTC 5 --> Read Receive Buffer

The data format is as follows;

Count	Data	Data		Data
-------	------	------	-------	------

Count Same as CTC 3

Data Same as CTC 3

CTC 6 --> Read Status Reg High

bit 0	-> Interrupt pending
bit 1	-> Receive mode
bit 2	-> Transmit mode
bit 3	-> NAK/Enq Error
bit 4	-> Undetermined Error
bit 5	-> Framing Error
bit 6	-> LRC Error
bit 7	-> Transmit Control Character

CTC 7 --> Read Status Reg Low

bit 8	-> Timeout Error
bit 9	-> Transmit Buffer Full
bit A	-> Receive Buffer Full
bit B	-> CTC 3 data overflow (count > hex 'FE')
bit C	-> Receive Control Character
bit D	-> Transmit Buffer Dropped
bit E	-> Receive Count Exceeded (count > hex 'FE')
bit F	-> not used

CTC F --> Read Card ID (hex '2B')

CTC 2	--> not used
CTC 4	--> not used
CTC 8	--> not used
CTC 9	--> not used
CTC A	--> not used
CTC B	--> not used
CTC C	--> not used
CTC D	--> not used
CTC E	--> not used

IBM Internal Use Only

• EPROM4-A

```

CTC 0  --> I/O Select
CTC 2  --> Write Eprom Data High Reg
CTC 3  --> Write Eprom Data Low Reg
CTC 4  --> Write Eprom Adr High Reg
CTC 5  --> Write Eprom Adr Low Reg
CTC 7  --> Write Control Reg
        bit 8  --> Group Select B0
        bit 9  --> Group Select B1
        bit A   --> Group Select B2
        bit B   --> Eprom Chip Enable
        bit C   --> Eprom Program
        bit D   --> Vpp Program
        bit E   --> Vcc Program
        bit F   --> Spare
CTC 8  --> Read Eprom Data High
CTC 9  --> Read Eprom Data Low
CTC A  --> Read Data High Reg
CTC B  --> Read Data Low Reg
CTC C  --> Read Adr High Reg
CTC D  --> Read Adr Low Reg
CTC E  --> Read Control Reg
CTC F  --> Read Card ID

CTC 1  --> not used
CTC 6  --> not used

```

SOFTWARE AVAILABLE FOR TPAC-A

• TPAC-A PROM PART NUMBERS

Auto-Poller Software

```

1.  PROM 1  ---> 6881491
2.  PROM 2  ---> 6881492
3.  PROM 3  ---> 6881493
4.  PROM 4  ---> 6881494
5.  PROM 5  ---> 6881495
6.  PROM 6  ---> 6881496

```

Pulsed Output Modulation, POM

```

1.  PROM 1  ---> 6881501
2.  PROM 2  ---> 6881502
3.  PROM 3  ---> 6881503
4.  PROM 4  ---> 6881504
5.  PROM 5  ---> 6881505
6.  PROM 6  ---> 6881506

```

Semiconductor Eqp Com Std, SECS

```

1.  PROM 1  ---> 8668897
2.  PROM 2  ---> 8668896
3.  PROM 3  ---> 8668899
4.  PROM 4  ---> 8668894
5.  PROM 5  ---> 8668895
6.  PROM 6  ---> 8668898

```

TEST CABLES (ALL CARDS)

IBM Internal Use Only

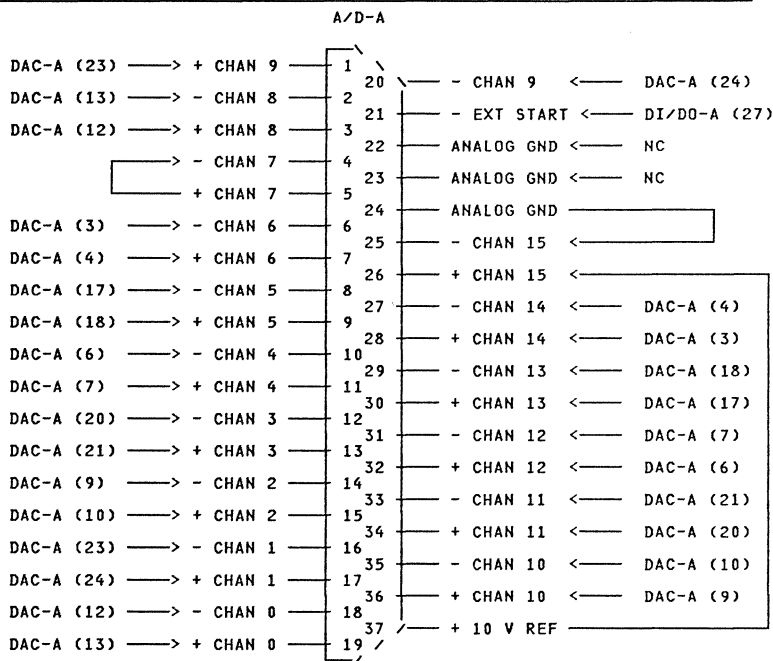


Figure 11. A/D-A Test Cable: Need 37 pin D-Shells for the DI/DO card, and the A/D-A card, and a 25 pin D-Shell for the DAC-A. All male type.

IBM Internal Use Only

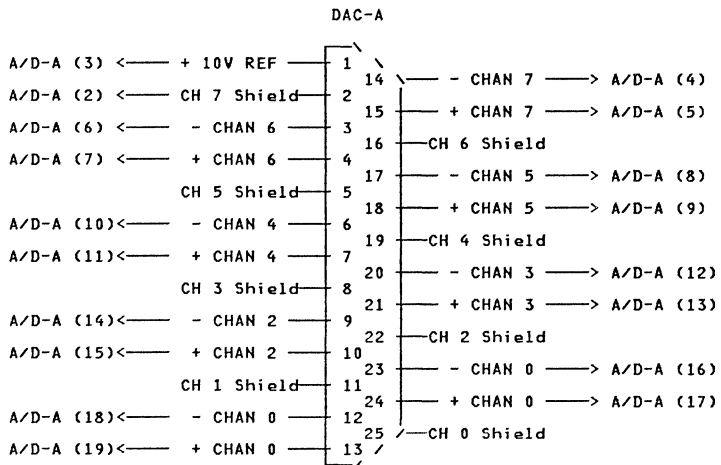


Figure 12. DAC-A Test Cable: Need 37 pin D-Shell for the A/D-A card and a 25 pin D-Shell for the DAC-A. All male type.

32 BIT DI/D0-A

DO BIT 15 (1)	————>	DI BIT 15 (30)
DO BIT 14 (20)	————>	DI BIT 14 (12)
DO BIT 13 (2)	————>	DI BIT 13 (31)
DO BIT 12 (21)	————>	DI BIT 12 (13)
DO BIT 11 (3)	————>	DI BIT 11 (32)
DO BIT 10 (22)	————>	DI BIT 10 (14)
DO BIT 9 (4)	————>	DI BIT 9 (33)
DO BIT 8 (23)	————>	DI BIT 8 (15)
DO BIT 7 (5)	————>	DI BIT 7 (34)
DO BIT 6 (24)	————>	DI BIT 6 (16)
DO BIT 5 (6)	————>	DI BIT 5 (35)
DO BIT 4 (25)	————>	DI BIT 4 (17)
DO BIT 3 (7)	————>	DI BIT 3 (36)
DO BIT 2 (26)	————>	DI BIT 2 (18)
DO BIT 1 (8)	————>	DI BIT 1 (37)
DO BIT 0 (27)	————>	DI BIT 0 (19)

Connect Digital Outputs to Digital Inputs
On The Same Connector

Figure 13. 32 Bit DI/D0-A Test Cable: Need only a 37 pin D-Shell for the DI/D0 card. Male type

IBM Internal Use Only

TPAC-A

DI BIT 0	(1)	————>	DO BIT 0	(22)
DI BIT 1	(14)	————>	DO BIT 1	(10)
DI BIT 2	(2)	————>	DO BIT 2	(23)
DI BIT 3	(15)	————>	DO BIT 3	(11)
DI BIT 4	(3)	————>	DO BIT 4	(24)
DI BIT 5	(16)	————>	DO BIT 5	(12)
DI BIT 6	(4)	————>	DO BIT 6	(25)
DI BIT 7	(17)	————>	DO BIT 7	(13)

**Connect Digital Outputs to Digital Inputs
On The Same Connector**

Figure 14. TPAC-A Test Cable For POM CODE: Need only a 25 pin D-Shell for the TPAC-A card. Male type

Heading ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Heading References</u>
comm	USER	1	Communications
pow	USER	16	24 Volt Power Connector

Figure ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Figure References</u>
switch	USER	3	1:
input	USER	4	2: 4
output	USER	5	3: 5
adjump	USER	6	4: 7
res	USER	7	5:
dacj	USER	9	6:
jump	USER	15	7: 15
24volt	USER	16	8: 16
io1	USER	18	9:
io2	USER	19	10:
ada	USER	28	11:
daa	USER	29	12:
dio	USER	29	13:
tpio	USER	30	14:

Imbed Trace

ARIEL I/O Unit Test System Manual

Control Systems Engineering

**IBM Corporation
General Technology Division
D-035 B-300-47A
East Fishkill
tel 533-7654**

IBM Internal Use Only

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

INTRODUCTION	1
Related Documentation	1
TEST HARDWARE SET UP	1
System Setup Cables	2
Test Cables (All Cards)	4
Ariel SYSTEM OVERVIEW	4
GETTING STARTED	6
Some Keyboard Conventions:	9
Loading Program From Disk	10
Copy Programs From Floppy To Fixed Disk	10
In Case of Test System Failures	10
SOFTWARE GENERAL SPECIFICATION	11
ATST	13
Main Menu	13
(AI) ANALOG INPUT CARD TEST	13
Program Description	14
AITA Screens	15
Sub-tests	18
Error Handling	21
Test Cable	24
Debug Procedure	25
(AO) ANALOG OUTPUT CARD TEST	27
AO/A Main Menu	27
Test Operation	28
DAC-A Utility Menu Options	29
Test Cable	31
Debug Procedure	32
(IO) DIGITAL INPUT/OUTPUT CARD TEST	34
Test Program	34
Main Menu Options	35
Test Cable	37
Debug Procedure	38
(DC) DESTINATION COMMUNICATIONS ADAPTER CARD TEST	40
Test Set-up	40
Block Address Test	40
Test Program	41
Control Tests	42
Operation Monitor Test	42
Interrupt Test	43
Repeater Function	44
Repeater Test Connections	44
Repeater Test	44
Bypass	45
Bypass Test	45
Diagnostics	45
Preliminary Tests	45
Input Voltages	46
Clock Oscillator	46
Data Input	46
Debug	46
(PO) PULSED OUTPUT MODULATION CARD TEST (TPAC-A)	48
Test Procedure	48
Test Cable	48
(RC) ROC CARD TEST (TPAC-A)	49
Test procedure	50
(MF) MFTJ	50
(BU) BURN IN TEST	50
(CL) CARD LIST	51
(RW) READ WRITE TEST	51
(PL) PLANAR BOARD & SUPPORT CARD TEST	55
General Information	55
Software	56
Main Menu	56
Debug Chart	59
SUPPORT CARD	59
Circuits to be tested	60
Procedure for required tests	60
Debug Chart	61

IBM Internal Use Only

Appendix A. Glossary	63
Appendix B. Software Libraries	65
DSC IV Software Library	65
ARIEL Software Library	65

IBM Internal Use Only

LIST OF ILLUSTRATIONS

Figure 1.	ARIEL Test Configuration	2
Figure 2.	Host (ECN/S1) to DSC IV cable	3
Figure 3.	Host (ECN/S1) to ARIEL Controller Cable	3
Figure 4.	Console to DSC IV Cable	3
Figure 5.	DSC IV to DISK Cables	4
Figure 6.	Ariel Mother Board	5
Figure 7.	I/O Card PNs for Ariel	6
Figure 8.	Front view of the DSC Box showing the DIS Maintenance Panel	7
Figure 9.	ATST Main Menu	13
Figure 10.	ADC-A test MAIN MENU	15
Figure 11.	AITA Screen Format	16
Figure 12.	ADC-A sub-test DIAGNOSTIC MENU	17
Figure 13.	ADC-A sub-test CHANGE LSA MENU	17
Figure 14.	ADC-A sub-test UTILITIES MENU	17
Figure 15.	A/D-A Test Cable	24
Figure 16.	ADC-A Test Procedure	26
Figure 17.	DAC-A Test Cable	32
Figure 18.	DAC-A Test Procedure	33
Figure 19.	DIO32-A test MAIN MENU	34
Figure 20.	32 Bit DI/DO-A Test Cable	37
Figure 21.	DIO32-A Test Procedure	39
Figure 22.	DCA-A Test Configuration	40
Figure 23.	Start of DCA card test	41
Figure 24.	DCA-A Test Cable	44
Figure 25.	TPAC-A Test Cable For POM CODE	49
Figure 26.	Card List Screen	52
Figure 27.	ERROR DISPLAY While RWTA is running	53
Figure 28.	RWTA Error Log	54
Figure 29.	RWTA Option Menu	55

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

INTRODUCTION

This manual provides the Test operator complete information to test and debug the I/O cards from the Ariel project or the RDIS Macro-Function Cards. All software related to this manual is available to run on the DSC IV or ARIEL Controller. The test philosophy, all associated hardware and software, and their setup procedures will be described to make the process as simple as possible. Specification references to all pertaining hardware and software is provided along with a general description of the function of the Hardware being tested.

This software and hardware can be used to test not only the Ariel I/O unit but also the RDIS I/O Cards. With software to test the Support card and mother board. The unit and all the I/O cards of the RDIS can be tested using the same procedures.

The software can be run on a DSC IV or ARIEL Controller, but this document will reference as a controller the DSC IV. And as the Macro-Function Cards to be tested this document will reference only the ARIEL I/O Unit with its Macro-Function Cards.

See the software libraries required for the test system in the appendices.

Any questions or requirements should be directed to Dept. 035, East Fishkill Facility.

Related Documentation

	Ariel User Manual
	Ariel Functional Specification
E45-1353-1/Doc PN 6951823	RDIS Functional Specification
E45-1289-0/Doc PN 7834982	Distributed Sensor Subsystem IV Functional Specification
Z45-0918,	Distributed Interface System, System Summary
Z45-1123,	Distributed Sensor Sybsystem User Guide Volume I
Z45-1124,	Distributed Sensor Sybsystem User Guide Volume II

TEST HARDWARE SET UP

The test hardware is that which is necessary to efficiently test and debug all the cards that plug into the ARIEL I/O Unit and the I/O unit itself.

The following is a list of the necessary hardware required for testing the ARIEL I/O Unit :

- DSC IV or ARIEL Controller
- S/I installed as an ECN with RS232 support
- RS232 gate for the ECN
- Coax cable (rg-62)
- Cable for ECN to DSC IV or ARIEL Controller communications (RS232)
- Cable for ROC to TPAC-A for Current-Loop Communications
- Wrap cables for I/O cards as defined in the Ariel User Guide
- Remote Operator Console (ROC)
- Oscilloscope
- DVM's

The hardware should be configured in the following manner:

IBM Internal Use Only

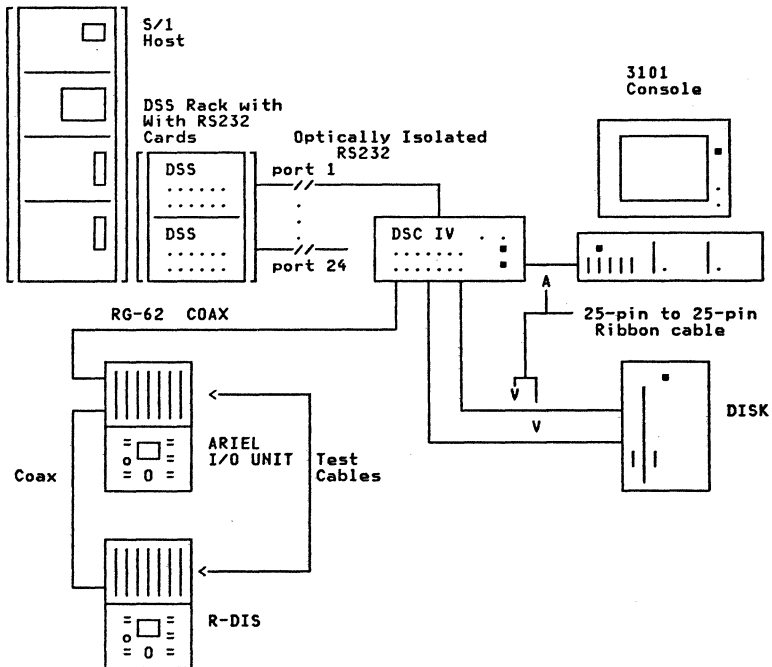


Figure 1. ARIEL Test Configuration

All the software can reside on the S/1 or Disk. It will be loaded to the DSC IV where it will be used to test the Ariel I/O cards and mother board. The link between the Host (S/1) and the DSC IV is shown in Figure 2 on page 3. The Disk is connected to the DSC IV via 25 pin D-Shells mass terminated. See Figure 5 on page 4 for all the details including what ports to connect to on the DSC IV. The only cable left is that which will allow communications to the Macro-Function Card which are plugged into the ARIEL I/O Unit. This cable is a single coax that is connected between the DCA of the ARIEL I/O Unit (the IN port) and J1 of the DSC IV. If another ARIEL I/O Unit is required the OUT port of the first ARIEL I/O Unit must be connected with another coax cable to the IN port of the next ARIEL I/O Unit. This can continue to a maximum of 32 ARIEL I/O Unit daisy-chained off each DSC IV.

System Setup Cables

IBM Internal Use Only

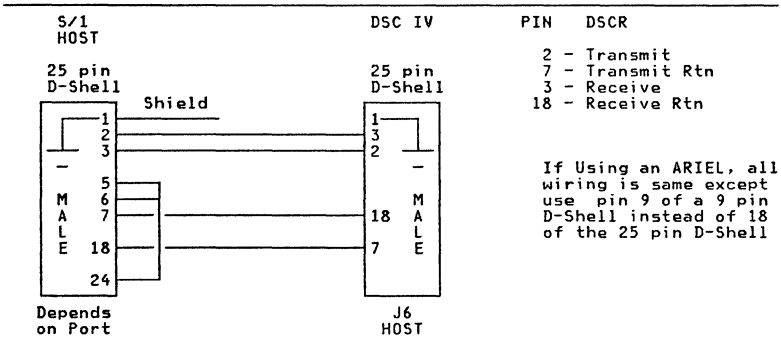


Figure 2. Host (ECN/S1) to DSC IV cable: For Test this cable should be a shielded twisted pair with stranded conductor

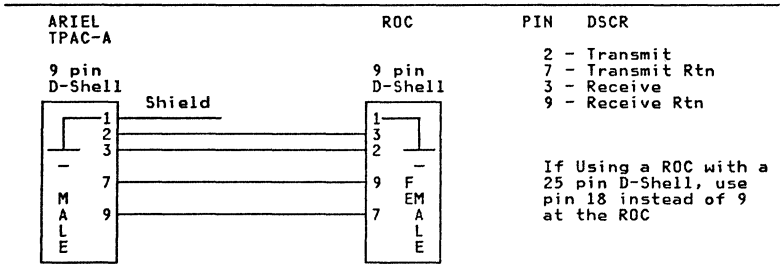


Figure 3. Host (ECN/S1) to ARIEL Controller Cable: For Test this cable should be a shielded twisted pair with stranded conductor

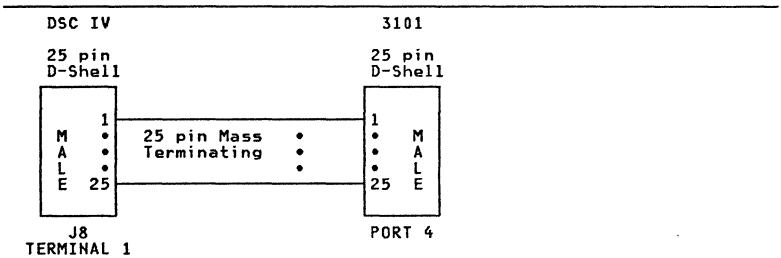


Figure 4. Console to DSC IV Cable: For Test this is a Ribbon cable and no shield is necessary

IBM Internal Use Only

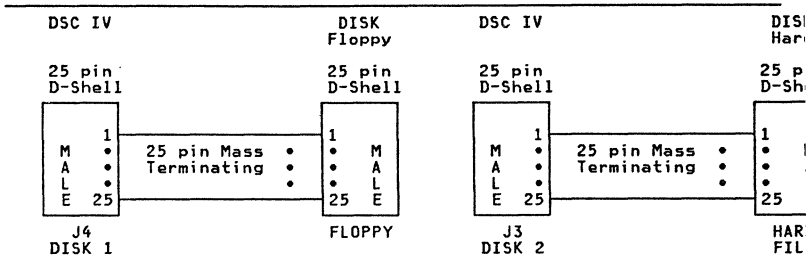


Figure 5. DSC IV to DISK Cables: For Test this is a Ribbon cable and no shield is necessary

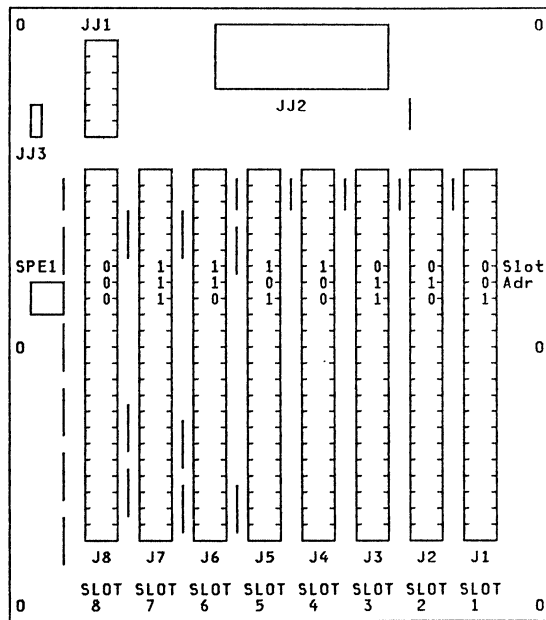
Test Cables (All Cards)

The Test System shown in Figure 1 on page 2 shows a RDIS and test cables that allow test of the I/O. For each I/O card, see Figure 7 on page 6, there is a test cable to wrap Outputs to Inputs. Each card has a different function thus a different test cable. To implement the test function a RDIS is necessary. For instance when testing the A/D-A (Analog to Digital converter for Ariel) a DAC-A (Digital to Analog Converter for Ariel) is necessary. To test an analog input signal an analog signal from the DAC-A must be generated. This RDIS can have any address (0-32) as long as it is not the same as the Ariel Test system. No two units on the same link (from the same DSC IV) can have the same address. The software allows changing the address at any time.

See the section on each cards test software in "ATST" on page 13, for the description of the test cables.

ARIEL SYSTEM OVERVIEW

The ARIEL I/O Unit is a very versatile box. Providing ease of use and function from simple DI and DO to complex horizontal processor control. This function is implemented on what are called Macro-Function Cards. These cards are plugged into the slots available in the ARIEL I/O Unit to generate the function required by the end user (because this is only test, we are not concerned with the configuration of cards but the ability to test efficiently). The design of Ariel allows any card any slot except in one instance, slot 8. This slot is reserved for the DCA (Destination Communication Adapter). It allows communications to the host (DSC IV or ARIEL Controller) via the DIS Channel through Coax Cable. See Figure 1 on page 2. In this case the host is a DSC IV but it could be an Ariel or a Personal Computer. Because of it's function, decoding the commands and presenting them to the correct &mfrc, this card must be placed in slot 8 of the mother board. The mother board consists of 8, 120 pin Amp connectors with miscellaneous r-pacs, c-pacs, an oscillator, and a 20 pin Amp connector. The slot which has the 120 pin and the 20 pin Amp connectors is slot 8. See Figure 6 on page 5.



COMPONENTS: J1-J8 120-pin A-BUS connectors
 JJ1 20-pin LSA/LSI connector
 JJ2 36-pin power connector
 JJ3 +24 vdc fan connector
 SPE1 system oscillator

Figure 6. Ariel Mother Board: Not shown are rpacs, and caps. The 20 pin connector distributes the card selects (LSA's) and the Interrupts from the cards.

IBM Internal Use Only

The following is a list of all the Ariel I/O cards to test/debug including the mother board:

CARD	ASM P/N	RC P/N	SCH P/N	EC P/N
DID032-A	6912757	6912758	6912759	A37493F
ADC-A	6912766	6912767	6912768	A37493F
DAC-A	6912769	6912770	6912771	A37493F
MOTHER BOARD	6912763	6912764	6912765	A37493F
TPAC-A	6912778	6912779	6912780	A37493K
DCA-A	6912784	6912785	6912786	A37493M

Note: These are the original released P/Ns. They could change due to EC's.

Figure 7. I/O Card PNs for Ariel

DID032-A - 32 bit digital input/digital output card
 16 DIs and 16 DOs for Ariel
A/D-A - Analog to Digital converter for Ariel
DAC-A - Digital to Analog converter for Ariel
TPAC-A - Thorpe Programmable Application Controller for Ariel
DCA-A - Destination Communications Adapter

GETTING STARTED

Power on the Test system in the following sequence:

1. Turn Disk power on (in back of the disk drive)
2. Turn 3101 power on (on front of control unit)
3. Turn DSC IV power on (in back of DSC IV where D-Shells are)

Now it is time to IPL the DSC IV. This is accomplished by turning the DBO ENABLE key in the front of the DSC IV to the ON position and then the OFF position, see Figure 8 on page 7.

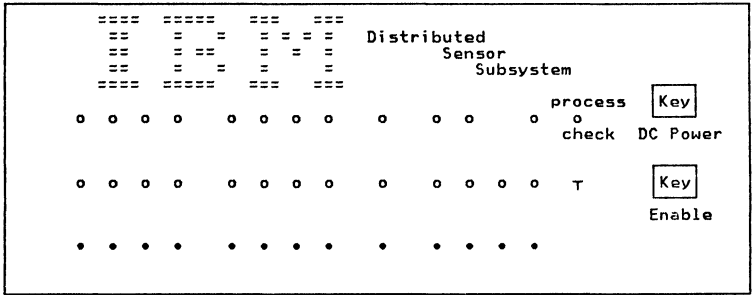


Figure 8. Front view of the DSC Box showing the DIS Maintenance Panel: o - LED, • - Toggle Switch, T - Push Button. If process check light is blinking, the controller has experienced unrecoverable error and the Test System should be reIPLed (by turning the Enable key).

IBM Internal Use Only

The Enable key, when on, enables the BIT switches on the front of the DSC IV. When off, the BIT switches are disabled and Hazel takes over the bus. Hazel is the name of the main processor of the DSC IV. After turning the DBO ENABLE key the DSC IV will try to down load the nucleus (this is the operating system for the DSC IV) from host Series/1. If the cable is disconnected or the S/1 is down, or the nucleus for the DSC IV is not on the S/1, the DSC IV will try 15 times, time out, and then try to load from disk. If the disk is down, or the cables off, or there is no nucleus on either the floppy or the hard file, the DSC IV will timeout and display on the screen;

DSC IPL FAILED

Verify the cables are connected correctly. Check that the S/1 is up and running properly, and also the Disk drive. If all this is O.K., then verify the nucleus for the DSC IV is on the S/1 and that is is correct (ie. verify the compile date which should be within the first few hundred words of the nucleus). Also check the floppy diskette in the disk drive (using the S/1) for the correct nucleus. One of these must be the problem. If the DSC IV still will not come up then consult D/035. If all goes well then follow the Yellow Brick Road to the next section.

If the system IPLs correctly the following will appear on the console (3101 screen).

```
*****
* EDXJ/4 : NUCLEUS LOADED *
***** 7/10/86 13.23 *****
```

Now it is time to load the Test program, configure the RDIS and ARIEL I/O Unit for test. Never ever ever plug in or pull out a RDIS Macro-Function Card or an Ariel Macro-Function Card while power is on.

IBM Internal Use Only

Some Keyboard Conventions:

These special control keys are listed below:

Special Control	Keys to Press	Purpose & Symbol used in this document
Attention	ALT and PF7 (Keypad on the right)	Enter System Commands and get attention of system. Symb: >
Reset	Reset	To clear the Lock-Mode/SetUp-Check condition (displayed on the bottom of the screen). Symb: (Reset)
Carriage Return	Send	Terminate keyboard entry. Symb: (Send)

IBM Internal Use Only

Loading Program From Disk

A load command is provided to load and start the execution of a program. For example, in order to load and start the execution of a program called MYPROG, the following command should be entered.

```
>$L GCODE (send)
A
A
      |
      |----- name of the program
      |
      |----- attention (ALT and PF7)
```

Copy Programs From Floppy To Fixed Disk

Test System Software and program updates will be sent to its users via floppy diskette. Users should transfer updates to the fixed disk upon receiving the floppy diskette using the disk utilities provided by the system software. The disk utility program has to be loaded into the system before it may be executed. The name of the disk utility program is DUT. To load and start the DUT, just enter the following on the keyboard:

```
>$L DUT (send)
```

The disk utility program will display to the operator a menu of different disk functions provided. Use the CP (Copy Program) command to transfer the new program to the hard disk.

In Case of Test System Failures

If the "Process Check" light started to flash on the DSC IV it means the microcontroller has experienced unrecoverable error. At which time, the Test System should be restarted by turning the the Enable key on the Maintenance Panel (ie. IPL).

If "Process Check" persists, DSC IV diagnostics should be executed to find out what is wrong with the processor.
DSC IV Test Spec, E45 1506 (PN 6159245) should be used along with the DSC IV diagnostics.

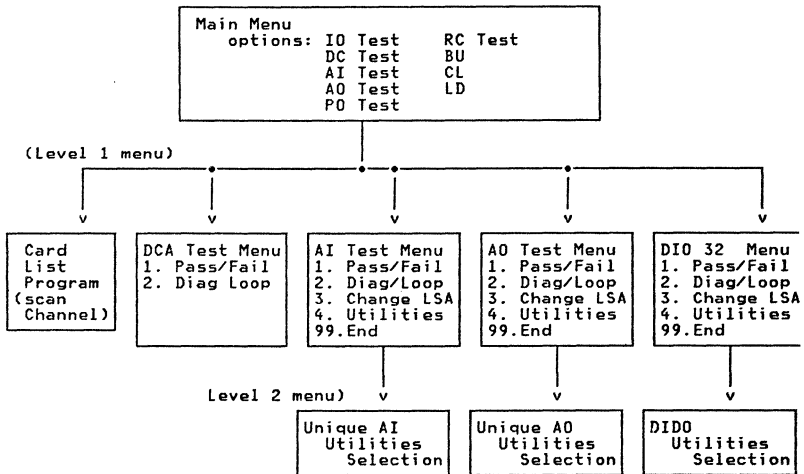
IBM Internal Use Only

SOFTWARE GENERAL SPECIFICATION

The Test System Software is a collection of programs for testing and debugging the Mother Board, the DCA-A, and the Macro-Function Cards. Other routines provide useful information such as scanning the DIS Channel displaying the card ID and its LSA found on the DIS channel, to the operator of the Test System.

A host program, ATST, presents a main option menu to the user so a card or board test may be loaded and executed. A logical structure of the Test System Software is presented below. If additional features are to be added in the future, it should be written so that it fits into the Test System Software architecture.

IBM Internal Use Only



Note: All the options of the main menu are not shown here.

Pass/Fail Test option should run a standard set of tests, determined by the engineer who designed the card/board, so if the card passes all the tests it may be installed in a system to run applications.

Diagnostic option should be used to debug cards that did not pass the Pass/Fail test. This option should guide the technician through the same steps of tests entered by the Pass/Fail option but stopping at each step, sometimes even enter a loop mode, to give the technician a chance to examine data output or measuring signal levels with an oscilloscope in order to pin point the source of errors.

Change LSA option gives the technician an opportunity to look at a similar card in the system and then switch back the original card.

Sometimes, other cards are used in conjunction with a card which is undergoing test; for example to simulate analog input signal for the Analog In(AI) card. Utilities give user the ability to set up these auxiliary cards.

Exit option is used to terminate the test program and it is usually option #99.

Programs written for the Ariel Test System should conform to the following conventions (and those written in the future should also follow the same conventions) in order to provide a simple operator interface for the Remote DIS Test System.

- Use alphabetic characters for choosing an option from the Main Menu.
- Main Menu display program will store the LSA of a selected card in the SYSCOM area(common storage space) so a test program should pick up LSA from SYSCOM.
- When fault is encountered, 3101 System Console will beep in order to attract the operators attention.
- Standard option menu should be used by all test programs.
 - Pass/Fail Test

```

- Diagnostic
- Change LSA
- Utilities
- Exit

```

- Test programs should display status information such as current card LSA and selected option on the top right corner of the 3101 display.
- Use >EL to terminate a looping sequence.
- In Pass/Fail test, no detail information about the failures encountered will be displayed. The error messages should be displayed under the diagnostic test option.

ATST is the main program of the test software. It is basically a menu that allows options to be selected for testing each of the I/O cards. When an option is selected, another program (the Test Program) specific to that option is loaded from host and a new menu will be printed on the console. This menu allows selecting of specific functions to test the card, (ie. Pass/Fail, Diagnostics, Looping, Single Step ... etc). These will be discussed later. Each card has it's own Test Program. These Test Programs have been written to be as uniform as possible, for ease of use.

```
*****  
                      IBM  
*****  
      REMOTE DISTRIBUTED INTERFACE  
      CUSTOMER INTERFACE CARD TEST  
*****  
-----MAIN MENU -----  
AI   ANALOG INPUT CARD TEST          CL   CARD LIST  
AO   ANALOG OUTPUT CARD TEST         FI   FIND CARD  
IO   DI/DO CARD TEST                 BU   CARD BURN IN  
DC   DEST COM ADPT CARD TEST         MF   MFTJ  
PL   PLANAR BOARD TEST(RDIS)        LP   LOAD PROGRAM  
PO   POM CARD TEST - TPAC-A         HP   HELP  
RC   ROC CARD TEST - TPAC-A         EN   END TEST PROGRAM  
RW   READ WRITE CARD TEST           ** ENTER OPTION ==> _
```

	option	description		option	description

Note: ATST, when loaded into DSC IV, must be loaded into partition 0 (address X'0000' to X'7FFF') because the Test Program's attention routines reside in ATST and they, the Test programs, only know to search Partition 0 when an attention key is entered. This allows the Test Programs to be loaded in any partition including partition 0.

AITA (Analog Input Test software for Ariel) will be used for the purpose of testing the functionality of Analog-Digital Converter cards

IBM Internal Use Only

(PN:6227044 for the RDIS or PN 6912766 for Ariel). The program is written in EDX/J to run on a DSC IV . This program is loaded into the DSC IV when option "AI" is selected from the main menu of ATST . See Figure 9. Some features of the program include:

- menu-driven execution
- 2 modes of operation - pass/fail and diagnostic
- ability to change macrofunction slots for test cards and test support cards
- takes appropriate actions on DISWRITE/READ errors
- allows for looping on a failure
- provides diagnostic text (via 3101 terminal) to aid the user in debug
- maintains a 'sub-test code' that allows testing to begin at any of the sub-tests that comprise the total test
- current ADC-A slot and action dynamically updated to screen
- audio alarm on any failure (except when looping)
- utilities to address any card/ any command

The following hardware is required for execution of AITA

- Ariel Test System
- ADC-A Test System Cable See Figure 15 on page 24.
- digital multimeter
- oscilloscope

Note: The ADC-A card being tested should be configured for +/-10V range. The DAC card in the system should be set for 0-5V range.

Note: The ADC-A card must be located in BLOCK 1 for Ariel and BLOCK 2 for DSC IV, although any macro is valid. Also the RDIS or the ARIEL I/O Unit must be powered up or AITA will not load for Ariel (DSC IV will run but problems can occur when running the interrupt test) This is due to interrupts.

Note: The DI/DO test card for Rdis must be EC'd as follows. Delete Z5(E01) from J1(27), and ADD TP10 to J1(27). This will guarantee the EXT START bit is longer than 250 nsec and shorter than 40 usec. If EC not in place, Sub-test code 'D' will fail occasionally.

Note: Sub-test code 'D' will fail for the original Rdis ADC, if the IBM A/D control module is used (PN 8691988). It should be replaced by PN 6256776 to resolve this problem.

The program utilizes 1 DAC from the RDIS (7 of the 8 channels) requiring that the Test System has the 7 channels wired to another slot for a total of 14 DAC signals to the ADC-A card. Also needed is a DIDO card from the RDIS. A single digital output is used to test the external start mode of the ADC-A.

Program Description

AITA is configured as a 2 part test program - pass/fail and diagnostics. The pass/fail test is designed as a fast-scan determination of the goodness of the card. The card is put through a series of sub-tests, either passing or failing. As soon as the card fails one of the sub-tests it

IBM Internal Use Only

fails the pass/ fail test. This is indicated with an audio alarm. For a card to pass the test it must pass all of the sub-tests.

The diagnostic test is designed for card debug and error correction. It executes the same sub-tests as the pass/fail test, but allows for looping on a test when an error occurs. The test can be started at the beginning with a check on test points or it can start at the sub-test indicated by the 'sub-test code'. This code can be carried over from the pass/fail test or can be changed by the user. See Figure 12 on page 17. Thus, if the pass/fail test is running and the card fails on a register sub-test, the diagnostic test can be entered at the register sub-test to loop on the command for debug purposes.

The diagnostic mode also allows for single-stepping of the sub-tests. This will allow the user to view the data screens for the sub-test before going on to the next sub-test. The failure of a sub-test is indicated by an audio alarm at which point the user must hit 'ENTER' to view the problem description.

The program also allows for changing the macrofunction slot for the ADC-A card to be tested or the test cards used by the program (DAC and D1D0 located in the RDIS). A utility is included that allows the user to do single or loop commands (any etc) on any card in the system as well as a convert loop for the ADC-A card.

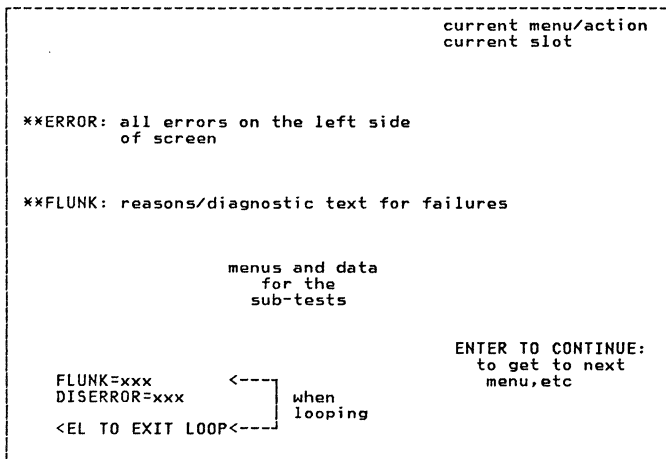
AITA Screens

The program menu and sub-menus are described below. These descriptions, in conjunction with the sub-test description and the debug procedures should be an adequate guide to using the program.

**** A/D CONVERTER TEST ****	
	MENU: LEVEL1 ADC LSA= 0000
** MENU OPTIONS:	
	1->PASS/FAIL 2->DIAGNOSTICS 3->CHANGE LSA (ADC,DAC,D1D0) 4->UTILITIES 99->END TESTING
** ENTER CHOICE: __	
Figure 10. ADC-A test MAIN MENU: When selecting option AI from the ATST Main Menu this screen will be written to the 3101 console.	

Figure 11 on page 16 details the layout of all the screens used in AITA. Examples are shown in Figure 12 on page 17 Figure 13 on page 17 and Figure 14 on page 17.

IBM Internal Use Only



**NOTE: Any data given as x'fXX' indicates that the 2 least significant digits are 'don't care' values and only the MSD is used in the sub-test.

Figure 11. AITA Screen Format

All data entry to the program is in a decimal format with the following exceptions:

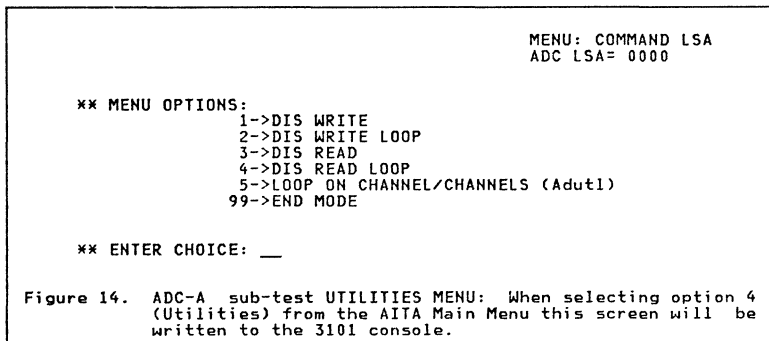
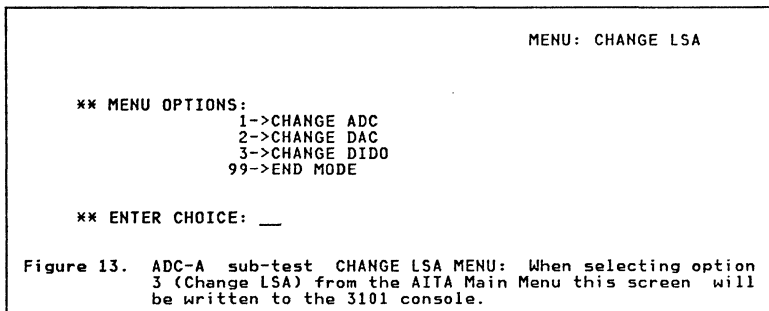
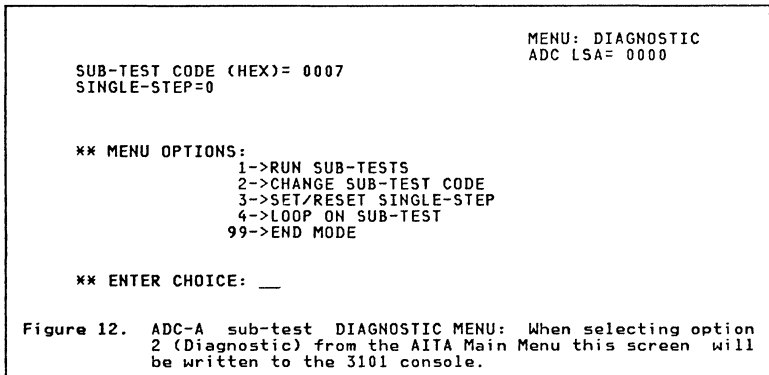
Sub-test code-> Enter in hex. Can be from 0 through C.

DIS Slot and CTC-> All entries for DIS data will be presented with the format required. For example:

BBM means to enter to hex digits for the block and one hex digit for the macro. The highest valid entry is 1F7.

BBMC is the same as above except the ctc is also entered as one hex digit. The highest valid entry is 1F7F.

IBM Internal Use Only



IBM Internal Use Only

Sub-tests

The following section describes the sub-tests currently used by AITA . FLUNK indicates the condition that fails the sub-test. A DIS error on the ADC-A card will also cause the sub-test to flunk. CHECK gives signals and modules that should be suspect for the failure. Refer to the ADC-A schematic while using this section for debug.

NAME-> Test-points SUB-TEST CODE-> 0 '0' **DIAG ONLY**

This test simply prints a number of test points to be checked along with the expected voltage. The test points are along the upper edge of the card. The points should be verified before any other sub-test is run.

FLUNK: a point is not at the specified voltage.

CHECK: the module originating the signal.

NAME-> INIT SUB-TEST CODE-> 1 '1'

DESC: performs a ctc1 and ctcD.

FLUNK: a DIS write error.

CHECK: LSA, SYNC and RETURN, command lines C0-C3, and module Z1, -PAR ERR should be high.

NAME-> EOC SUB-TEST CODE-> 2 '2'

DESC: performs a ctc2 and checks that bit0 is '1'. This bit is from the A/D converter module(Z16) and is active when a conversion completes. This signal is labelled +CONVERTING(same as -EOC) on the schematic.

FLUNK: bit0 of ctc2 is not = 1.

CHECK: +CONVERTING(Z16-20), control lines from Z1 to Z2, and driver Z8.

NAME-> HI REG SUB-TEST CODE-> 3 '3'

DESC: ripples a '1' through the hi reg(used for channel and gain selection) and then ripples x'55' through to the left. The writing uses ctc4 and verification is done with ctc6.

FLUNK: ctc6 data read not equal to ctc4 data written.

CHECK: receiver Z19, control lines from Z1 to Z2, driver Z8, +/- ENABLE DRV from Z1 to Z8, and parity check. Use data written and read as a debug aid.

NAME-> LO REG DIAG SUB-TEST CODE-> 4 '4'

DESC: puts the card into diagnostic mode using ctc9 and then ripples x'01' and x'55' through the lo reg using ctc5. The data is read back using ctc7.

FLUNK: ctc7 data read not equal to ctc5 data written

CHECK: same as for sub-test 3.

IBM Internal Use Only

NAME-> LO REG

SUB-TEST CODE-> 5 '5'

DESC: writes several patterns to the A/D Control module using ctc5 and verifies the resultant status with ctc7. Ctc4 is executed at the start to guarantee that the gain is set to 1. The card is reset after each ctc7 in order to clear the status. The following data is written and expected:

ctc5	ctc7
x'80'	x'C0'
x'C0'	x'E0'
x'20'	x'50'

FLUNK: ctc7 data read not what was expected.

CHECK: clock Z21, ctc5 bits to Z1, status bits from Z1 to driver Z8, -CONVERTING (Z16-17), and Z1 functionality.

NAME-> LO REG RESET

SUB-TEST CODE-> 6 '6'

DESC: writes the same data as LO REG test except the card is immediately reset and a ctc7 is then used to verify that the control module(Z1) was reset. The following data is written and expected:

ctc5	ctc7
x'80'	x'40'
x'C0'	x'40'
x'20'	x'40'

FLUNK: ctc7 data read not what was expected.

CHECK: same as sub-test 5.

NAME-> +10V REF

SUB-TEST CODE-> 7 '7'

DESC: channel 15 is converted and the result is checked for magnitude of x'7xx' and sign of '0'. The conversion is initiated with ctc4/5. The test polls for an interrupt pending(ctc7, bit0). After the interrupt pending is read the value is checked for magn/sign.

FLUNK: magnitude not equal to x'7xx', sign not equal to '0' or interrupt pending not detected after ten reads of the status register.

CHECK: +10V signal from channel 15 into Z16-30, 12 bit result from Z16 to Z17 and Z18 and from Z17 and Z18 to Z2, mux select lines.

NAME-> ZERO REF

SUB-TEST CODE-> 8 '8'

DESC: channel 7 is converted and the result is checked for magn/sign of x'0xx' or x'Fxx. The conversion is initiated with ctc4/5. The test polls for an interrupt pending(ctc7, bit0). After the interrupt pending is read the value is checked for magn/sign.

FLUNK: magn/sign not equal to x'0xx' or x'Fxx', or interrupt pending not detected after ten reads of the status register.

CHECK: same as sub-test 7 except look for 0V on channel 7.

NAME-> CHANNEL CHECK SUB-TEST CODE-> 9 '9'

DESC: +5V is written to the DAC outputs using ctc2 and x'FFFn' where n goes from 0 to 7. The corresponding ADC-A inputs are then converted(see cable layout). For each DAC channel written two ADC-A channels will be read, one will read +5V, the other will read -5V. Converted data should be x'4XX' for +5V and x'BXX' for -5V.

IBM Internal Use Only

FLUNK: magnitude equal to the 10V or 0V references, sign bit not correct (0 for +5V, 1 for -5V), or int pending not detected after 10 reads of the status register.
CHECK: same as sub-test 7 except look for +/-5V on the appropriate channel.

NAME-> INTERRUPT SUB-TEST CODE-> 10 'A'

NAME-> EXTERNAL START SUB-TEST CODE-> 11 'B'

DESC: the ADC-A card and D0 bit are reset and the ADC-A card is put into external mode using ctc5. The D0 bit is toggled acting as an external start (to start the conversion) and the status reg is interrogated for int pending.
FLUNK: int pending not detected after 10 reads of the status register.
CHECK: EXTERNAL START(Z1-J2) from D shell, Z1 functionality.

NAME-> CHECK GAIN SUB-TEST CODE-> 12 'C'

DESC: the DAC output and gain for the ADC-A are varied such that DACxgain=+5V. The testing is done using channel 0 of the ADC-A and the result is checked for accuracy with the 3 MSBs. The result(ctc2,3) should be x'4XX'. The following data is used for the test:

DAC data	gain(ctc4)
FFF	1
7FF	2
350	5
18F	10

FLUNK: the result of the conversions are not equal to x'4xx', int pending is not detected after 10 reads of the status reg.
CHECK: gain select lines from Z1 to Z14, Z14 functionality. If the conversion yields x'3XX' or x'5XX' vary R6 until the result corrects to x'4XX'.

NAME-> CHECK MUX SUB-TEST CODE-> 13 'D'

DESC: Uses the external start to convert channel 15 (+10v) and channel 7 (0v) to verify the operation of mux switching. This test will fail if Z1 is back level. The test fails if the 2 converted results are equal or the +10v channel converts to X'000' or X'F80'.

IBM Internal Use Only

Error Handling

See the description of the individual sub-tests as an aid to debug.

The following table lists some of the control lines that can be verified during debug:

ctc	-XR3	-XR4	-READ	-LOAD	-EN DRV	STATUS
ctc2	1	1	0	1	0	0
ctc3	1	0	0	1	0	0
ctc4	0	1	1	0	1	0
ctc5	0	0	1	0	1	0
ctc6	0	1	0	1	0	0
ctc7D	0	0	0	1	0	0
ctc7	0	0	1	1	0	1
ctc15	0	0	1	1	0	0

CHANNEL	CTC4 BITS	PINS ON Z10-Z13 4 1 16	PIN 3 (Z13,Z11)	PIN 3 (Z12,Z10)
0	1111	0 0 0	1	0
1	1110	0 0 1	1	0
2	1101	0 1 0	1	0
3	1100	0 1 1	1	0
4	1011	1 0 0	1	0
5	1010	1 0 1	1	0
6	1001	1 1 0	1	0
7	1000	1 1 1	1	0
8	0111	0 0 0	0	1
9	0110	0 0 1	0	1
10	0101	0 1 0	0	1
11	0100	0 1 1	0	1
12	0011	1 0 0	0	1
13	0010	1 0 1	0	1
14	0001	1 1 0	0	1
15	0000	1 1 1	0	1

IBM Internal Use Only

Flunk Messages

FLUNK messages are given any time a sub-test fails. If the test is in pass/fail mode a short error report is given. If the test is in diagnostic mode the error report and possible causes are given. A DIS error will cause the sub-test to flunk also. The messages are indicated with the heading 'FLUNK'.

Error Messages

The following lists and describes the possible error messages given by AITA:

ADC-A ID(CTC=F) TESTED BAD->

Occurs when a bad ID or DIS error are encountered when trying to read the ID of the current slot. This is done before entry to the pass/fail or diagnostic tests. The user is given the opportunity to loop on the slot, change the slot or return to the level1 menu.

PROBABLE CAUSE:

- .slot defined incorrectly
- .card not generating RETURN (Z1)

BAD ID READ AT NEW SLOT->

Occurs when a bad id is read when changing slots using option 3 of the level1 menu (CHANGE SLOT). Again the user is given the opportunity to loop on the slot, change the slot or end the mode.

PROBABLE CAUSE:

- .slot defined incorrectly

DISERROR ON CTC=F FOR NEW SLOT->

Occurs when a DIS error is encountered when changing slots using option 3 of the level1 menu. The user is given the opportunity to loop on the slot, change the slot or return to the level1 menu.

PROBABLE CAUSE:

- .card not generating RETURN (Z1 for the ADC-A)

DISERROR: CTC= n ->

Indicates the occurrence of a DIS error for any sub-test or subroutine used by a sub-test when dealing with the Ariel ADC-A card. The condition will not terminate a loop, however, it acts as a 'FLUNK' to the pass/fail and diagnostic mode. Thus, in pass/fail the mode is terminated, in diagnostics the loop option can be entered. 'n' is the ctc that failed.

PROBABLE CAUSE:

- .bad parity check(Z2,Z19) inhibiting RETURN

DISERROR ON ADC-A->

Indicates a DIS error while trying to reset the ADC-A card. Execution returns to the level1 menu.

PROBABLE CAUSE:

- .no RETURN (Z1) - enter DIAG mode

DISERROR ON DAC->

Indicates a DIS error when working with the DAC

IBM Internal Use Only

card. Execution automatically goes to the level1 menu.

PROBABLE CAUSE:

.bad card - notify D/035

DISERROR ON DID0->

Indicates a DIS error on the DI/D0 card. Again, execution goes back to the level1 menu.

PROBABLE CAUSE:

.bad card - notify D/035

INT PEND(EOC) NOT GOING ACTIVE->

Indicates that an End-of-Convert signal was not received from the A/D Converter module(Z16) after a conversion was started. The wait time is 10 DISREAD statements.

This error may occur in a number of sub-tests(any that verify conversion operations).

PROBABLE CAUSE:

.START CONVERT not generated (Z1,Z16,Z21)

LO REG NOT = X'40' AFTER RESET->

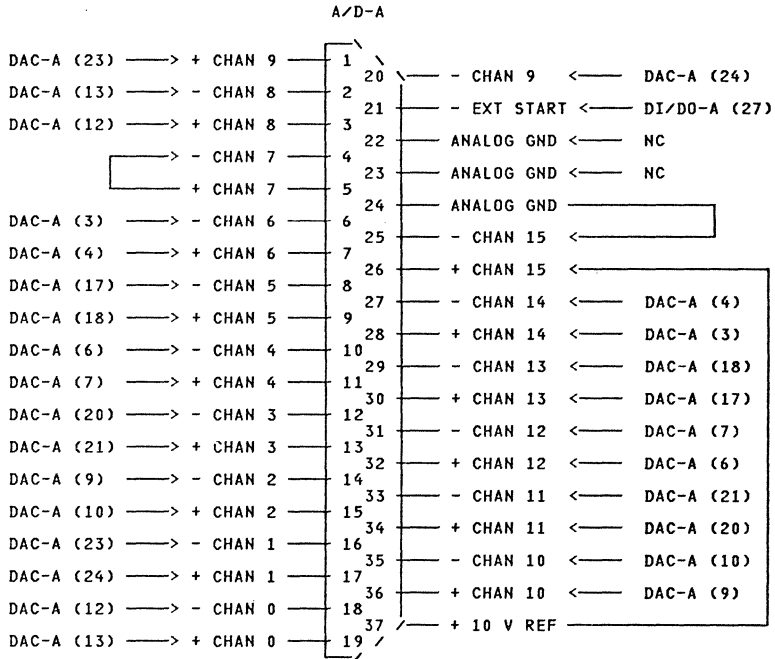
Indicates that an attempt to reset the ADC-A card failed. Execution goes to the level1 menu.

PROBABLE CAUSE:

.Z1 not resetting (z1,Z4) - enter DIAG mode

IBM Internal Use Only

Test Cable



The Number in parenthesis is the pin number to connect to.
ie. DAC-A pin (13) is connected to A/D-A pin 19

B/M for building ADC-A Test Cable
Amp P/N Description

205210-1 (X1)	J1	37 pin male D-Shell for mate to ADC-A
205210-1 (X1)	J2	37 pin male D-Shell for mate to DI032-A
205208-1 (X1)	J3	25 pin male D-Shell for mate to DAC-A
66506-9 (X65)		male pin
205731-1 (X2)		strain relief for 37 pin D-Shell
205718-1 (X1)		strain relief for 25 pin D-Shell
		Size 22 stranded Cable

Note: Use Crimper AMP PN 90302-1 or 90312-2 (RED DOT).

Figure 15. A/D-A Test Cable: Need 37 pin D-Shells for the DI/D0 card, and the A/D-A card, and a 25 pin D-Shell for the DAC-A. All male type.

IBM Internal Use Only

Debug Procedure

- Place the ADC-A card to be tested into an open slot of the Ariel I/O Unit.
- Connect test cable between ADC-A (Ariel I/O Unit) and DAC and DID032 cards (RDIS).
- Select AI(Analog In) Test on the system main menu.
- Follow the flowchart in Figure 16 on page 26.

IBM Internal Use Only

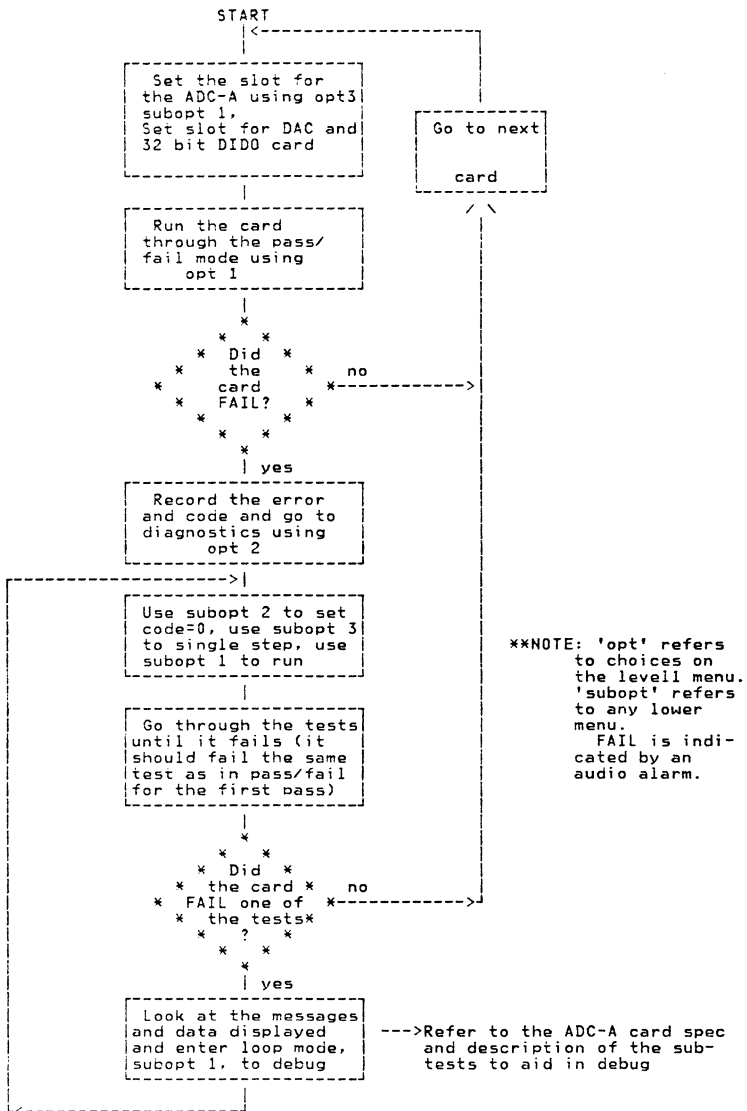


Figure 16. ADC-A Test Procedure

IBM Internal Use Only

(A0) ANALOG OUTPUT CARD TEST

This program tests all card functions of the Remote DIS DAC card (PN 6227043 and the Ariel DAC-A card (PN 6912769). This test has two modes, Pass/Fail and Diagnostic. Also provided is a menu of DAC-A Utilities which are designed to help exercise and debug the card. The program is written in EDX/J to run on a DSC IV. This program is loaded into the DSC IV when option "A0" is selected from the main menu of ATST. See Figure 9 on page 13.

The following hardware is required for execution of AOTA

- Ariel Test System
- DAC-A Test System Cable See Figure 17 on page 32.
- Oscilloscope
- Digital Multimeter

AOTA Main Menu

Following is the main menu for AOTA The options are described below.

MENU: MAIN
DAC LSA= 0000

**** DACTEST ****

SELECT TEST
1 = PASS / FAIL TEST
2 = DIAGNOSTIC TEST
3 = CHANGE LSA
4 = DAC UTILITIES
99 = END DAC TESTS

==>__

Pass /Fail Test

Selecting this option begins the pass/fail test of the DAC-A card. The pass/fail mode only requires the operator to plug in the card under test and connect a cable from it to an ADC card in the test rack. If a fault is found on the card, a short message is printed indicating only the particular test which failed. This test takes less than 5 seconds to complete. See "Test Operation" on page 28 for details of the program's test procedure.

Diagnostic Test

This option will begin the card test in diagnostic mode. In diagnostic mode, the card is subjected to the same tests as in pass/fail mode. The difference is that when a failure occurs the program issues a summary of the errors, suggests the area of the card which may be at fault and suggests which DAC-A Utilities would be most helpful in debugging the problem which has occurred. The DAC-A Utility menu is automatically entered at this point.

IBM Internal Use Only

Change LSA

This option allows the user to change the LSA (or slot) of the DAC-A to be tested. After selecting this option, the program will display the current LSA and prompt for a new one. By simply hitting the SEND key, the current LSA definition is retained. To change the LSA, enter a new Block and Macro address. For example, to select Block 'A', Macro '3'; enter 'A3'. To select Block '4', Slot '6'; enter '46'.

After a new LSA is selected, the program checks to see if there is a DAC-A in the selected slot. If the address does not respond or if the ID received is incorrect, the program will ask whether to proceed with this slot or select another. If this is actually the desired slot and there is just no card in it yet or if the card in the slot may have bugs which could have kept it from responding properly, then proceed with this address. If the incorrect slot was selected, then it may be changed.

The program will next ask for the new address for the ADC. During card test and debug there should be no reason to change the ADC LSA. By just hitting the SEND key, the ADC LSA will not be changed. If for engineering purposes there is a reason to change the ADC LSA, enter the new LSA in the same manner as the DAC-A LSA.

DAC-A Utilities

This option transfers the program directly to the DAC-A Utilities Menu. For details on the utilities, see "DAC-A Utility Menu Options" on page 29.

Test Operation

The test program itself is divided into five subtests which are performed in the order listed below.

- Register Test

This test checks the internal registers on the card by writing various patterns to the card's holding registers and then reading them back. This test also checks the control circuitry by insuring that all valid CTC's are executed properly. This test traps and logs all register errors or DIS errors which may occur. In diagnostic mode the program prints a summary of the errors, indicating what errors occurred and which CTC's caused them.

- Illegal CTC Test

This test tests the control circuitry by writing illegal CTC's to the card and insuring that the card does not generate a return in response. In diagnostic mode the program will indicate which CTC's received a response.

- Voltage Reference Test

This test checks the +10 volt reference on the card. If this reference is not correct the DAC-A's on the card cannot function properly. The reference signal is routed to channel 8 on the ADC. In diagnostic mode the program indicates the voltage limits for the reference and the actual voltage read.

- Analog Range Test

This test insures that all analog channels are within range and that the channel select circuitry is working properly. Each channel is routed to the ADC channel of the same number and then the maximum and minimum output values for each channel are tested. This card is built with the voltage range select jumpers in the 0-5 volt positions.

IBM Internal Use Only

However, since the card does not have to be calibrated for this test, the voltage may be out of this range. Therefore, the test checks to see that the values read are within the maximum possible range allowed by the pots with the jumpers in the 0-5 volt position. In diagnostic mode the boundary values are listed as well as a summary of the out of range voltages.

- **Analog Repeatability Test**

This test checks to insure that all analog values are repeatable. All channels are taken to their minimum and maximum values again as in the range test and these values are compared to the values seen in that test. If the second reading is not within a specified range from the first value then an error is flagged. In diagnostic mode the repeatability specification is listed along with both voltages read on the channels which failed.

DAC-A Utility Menu Options

MENU: UTILITY
DAC LSA= 0000

*** DAC TEST UTILITIES ***

1. COMMAND SINGLE STEP OR LOOP
2. REGISTER TEST
3. ILLEGAL CTC TEST
4. DAC TO ADC WRAP
5. ANALOG SCOPE LOOP
6. ANALOG REPEATABILITY TEST
7. CHANGE LSA
99. RETURN TO MAIN MENU

SELECT OPTION ==>__

The following tests are provided for debugging specific problems and exercising specific card functions. When a card fails in diagnostic mode, the program will suggest which utilities may provide the most help. In addition the user may select the Utilities Menu directly from the Main Menu.

Note: The following items apply to all of the utilities.

- **Standard Attention Functions**

Most of the utilities at some point, enter a continuous loop. To exit from a loop the following attention functions are available:

- >EL - End Loop and restart the current utility.
- >M - Return to the Utilities Menu.
- >EU - End Utilities and return to the Main Menu.

- **Calibration**

If the DAC-A card has not been calibrated, the voltages which the utility says it is sending will not be the actual voltages produced and read back from the ADC. The voltages produced should, however, be proportional to the desired voltages. If there is a need to calibrate the card, see the DAC-A Functional Specifications.

IBM Internal Use Only

- Floating Point Numbers

All voltages asked for by the utilities must be entered with a decimal point. For example 2 volts must be entered as '2.0'. Zero volts, however, may be entered as '0'.

Command Single Step or Loop

This option allows the user to issue any command to the card and get a response back (on a read). There is also an option which will allow user to loop on any command. This utility is useful for probing the control circuitry. This is the only utility which does not exit its loop when a DIS error occurs. This utility is sometimes useful for other tasks such as register checking and analog output manipulation.

The first prompt from this utility is for a command. Any command (0-F) may be issued. See the DAC-A Functional Specification for an explanation of the commands. If '99' is entered as a command the program returns to the Utilities Menu. Next the program will ask if it should loop on this command. If the selected operation is a write command the program will prompt for the data. After this the program issues the command and prints any data resulting from a read operation. If a loop was specified this operation continues until ended with an attention function. If no loop was specified the program asks for a command again.

Register Test

This utility is a loop which writes data patterns to the card using both CTC 1 and CTC 2,3. This test is designed to check the holding registers and their controlling circuitry.

When this option is selected, the program immediately begins a loop which writes data patterns to the card. If an error occurs, the loop stops and displays both the data written and the incorrect data read. At this point the program allows the user to decide whether to loop on the data pattern which caused the error or to resume the test with the next data pattern in the sequence. If no error occurs, the screen remains blank except for a counter in the lower lefthand corner which indicates the number of loops successfully completed. Use the attention functions to exit from this utility.

Illegal CTC Test

This routine is a loop which issues CTC's to which the DAC-A card is not supposed to respond and checks to be sure that it does not. This routine may be helpful in debugging control circuit problems.

When this utility is selected, the program begins a loop which issues all of the illegal CTC's to the card. If no errors occur the screen remains blank except for a counter in the lower lefthand corner which indicates the number of loops successfully completed. When an error occurs the user may loop on the CTC which caused that error or continue the test. To exit this loop, use the attention functions.

DAC-A to ADC Wrap

This routine allows the user to output voltages from the DAC-A and then read them back through the ADC. This is the most useful routine for analyzing analog problems. The routine has options which allow the operation to be performed continuously or only a specified number of times. The operation may be either a repeated writing of the same voltage or a

IBM Internal Use Only

staircase type function with the maximum, minimum and incremental voltages specified.

When this routine begins, it first asks for the number of the channel to be exercised and the voltage range as selected by the board jumpers. Next, the program prompts for the number of times the wrap test is to be executed. For a continuous loop, enter '-1'. The next question asks whether the wrap test should consist of repeated wraps with the same voltage or of a step type function. If a repeat wrap was selected the program will prompt for the desired voltage. If a step wrap was selected, the program will prompt for the minimum and maximum voltages and for the voltage by which the output is to be incremented each time. Use the attention functions to exit this loop.

Analog Scope Loop

This routine is a loop which produces either a squarewave or step function output from the card which is fast enough for a good scope display. This utility is helpful when checking the analog portion of the card with an oscilloscope.

The first prompt in this routine, asks if the output should be in squarewave or step function form. The program then asks for the channel number and jumper selected voltage range. Next, the routine asks for the minimum and maximum output voltages. If a square wave has been selected, the program will ask for the value of 1/2 of the square wave period (1/2 of the period is the width of the pulse) and then begins the loop. If the step function was chosen, the program will ask for the voltage corresponding to the height of each step and then begin the loop. Use the attention functions to exit this loop.

Analog Repeatability Test

This routine repeatedly writes and reads the specified voltage on the specified channel and then reports the average, maximum and minimum voltages read. This test allows a large number of readings can be taken and summarized so that a channel with excessive noise can easily be identified.

First the routine will ask for the channel number and the jumper selectable voltage range. Next it will prompt for the test voltage and the number of iterations. The test can perform 10,000 iterations in about one minute. This will give a reliable reading. When the test is complete, the program will summarize the results displaying the average, maximum and minimum voltages. Normally, 20 to 40 mV of noise can be seen between the test system and the RDIS box.

Change LSA

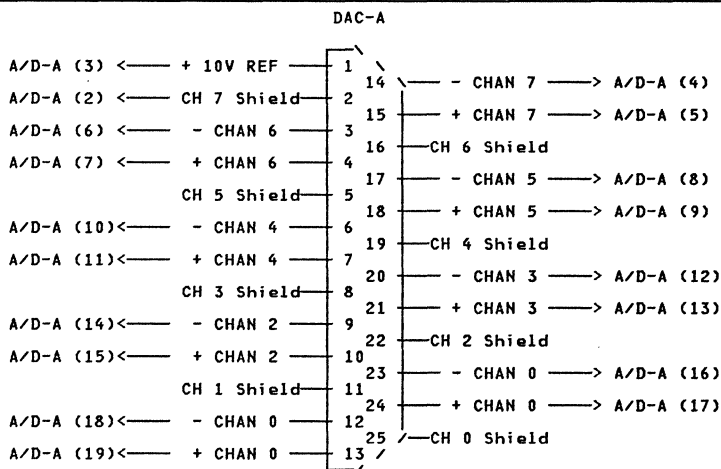
This option is exactly the same as the Change LSA option in the Main Menu. For details, see "AOTA Main Menu" on page 27.

Return to Main Menu

This option returns the program to the Main Menu.

Test Cable

IBM Internal Use Only



The Number in parenthesis is the pin number to connect to.
ie. A/D-A pin (19) is connected to DAC-A pin 13

B/M for building DAC-A Test Cable Amp P/N Description

205210-1 (X1)	37 pin male D-Shell for mate to ADC-A
205208-1 (X1)	25 pin male D-Shell for mate to DAC-A
66506-9 (X65)	male pin
205731-1 (X1)	strain relief for 37 pin D-Shell
205718-1 (X1)	strain relief for 25 pin D-Shell
	Size 22 stranded Cable

Note: Use Crimper AMP PN 90302-1 or 90312-2 (RED DOT).

Figure 17. DAC-A Test Cable: Need 37 pin D-Shell for the A/D-A card and a 25 pin D-Shell for the DAC-A. All male type.

Debug Procedure

- Place the DAC-A card to be tested into an open slot of the Ariel I/O Unit.
- Connect test cable between DAC-A (Ariel I/O Unit) and ADC card of the (RDIS).
- Select AO(Analog Out) Test on the system main menu.
- Follow the flowchart in Figure 18 on page 33.

IBM Internal Use Only

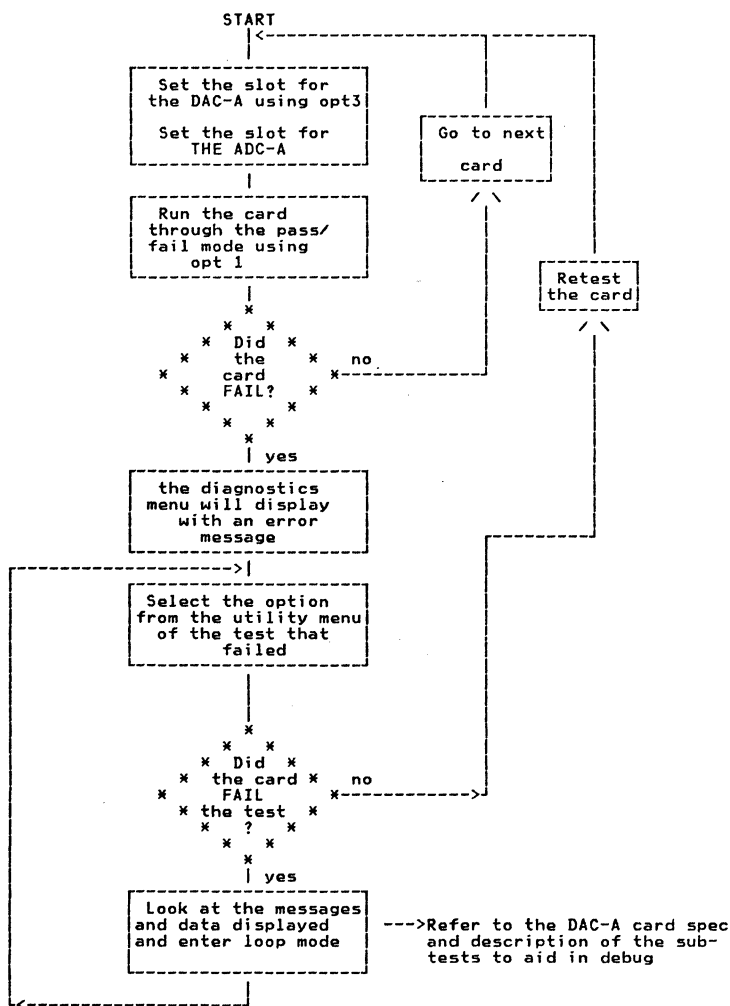


Figure 18. DAC-A Test Procedure

IBM Internal Use Only

(IO) DIGITAL INPUT/OUTPUT CARD TEST

IOTA (Digital Input/Output Test software for Ariel) will be used for the purpose of testing the functionality of Analog-Digital Converter cards (PN 6227041 for the RDIS and PN 6912757 for Ariel). The program is written in EDX/J to run on the DSC IV . This program is loaded into the DSC IV when option "IO" is selected from the main menu of ATST . See Figure 9 on page 13.

Test Program

Once the 32 bit DI/DO test program is initiated an option menu will appear as shown in Figure 19.

```
          32 BIT DI DO TEST           date
LSA = 0000
ENTER 1-> PASS/FAIL MODE
      2-> DIAG LOOP MODE
      3-> CHANGE LSA MODE
      4-> UTILITY (SINGLE STEP) MODE
      99-> TO END TEST
```

Figure 19. DI032-A test MAIN MENU: When selecting option IO from the ATST Main Menu this screen will be written to the 3101 console.

The test procedure of IOTA is described below.

1. A special 37 pin D-Shell connector cable wraps the DO to DI
2. Diagnostic data is written to the digital output register via CTC A,B
3. The digital output is read back via CTC 6,7
4. The digital input is read via CTC 4,5
5. The read digital output is compared with the Diagnostic data written
6. The read digital input is inverted and compared with the Diagnostic data written
7. If an error occurs it is flagged and written to the 3101 screen

Note: The diagnostic data (written to the DO) is generated as follows;

- 1) the diagnostic data is set to X'0000' and a B'1' is shifted through.
- 2) then the diagnostic data is set to X'FFFF' and a B'0' is shifted through.

IBM Internal Use Only

Main Menu Options

Single Step Mode

This mode allows diagnostic data to be tested one byte at a time each time enter is hit.

To start this mode type 4 at the main menu and hit enter the following screen will appear;

```
Current menu/option
Current Slot

MODE = SINGLE STEP
ATTN EL TO STOP LOOP

WRITE DATA DO  X'  '  B'  '
READ DATA DO  X'  '  B'  '
READ DATA DI  X'  '  B'  '
ERROR                      X (bad data bit)

ENTER TO CONTINUE
x   TO RESTART
d   TO GO TO DIAGNOSTIC LOOP
```

To get to the main menu -- >X or just X and hit enter.

Typing d and hitting enter will go to the diagnostic screen. See the chapter on the diagnostic mode for further details.

Pass/Fail Mode(Stop on Error)

This mode will allow a complete test of the 32 Bit DI D0 card. It will run all the diagnostic data and will stop only for an error. This mode will allow the user to select the number of loops the diagnostic data is tested.

Note: It takes approximately 5 seconds to complete 25 loops.

To enter this mode enter 1 at the main menu and hit enter. The following prompt will occur;

```
ENTER LOOP CNT ' ' ** DECIMAL NOTE 25 = 5 sec
NOTE ATTN EL TO STOP
```

The following errors can happen;

- 1) error on DIS write internal card problem
 - a) check for return while in diagnostic mode
 - b) check to see if ctc's are being decode properly
 - c) check for parity at parity checker
- 2) error on DIS read internal card problem or I/O wrap problem
 - a) check for return while in diagnostic mode

IBM Internal Use Only

- b) check to see if ctc's are being decode properly
- c) check for parity at parity checker

3) error reading card ID usually an LSA problem

- a) if LSA is wrong a prompt to enter new LSA appears
- b) enter new LSA

If an error occurs the following screen will appear;

```
Current menu/option
Current Slot

MODE = STOP ON ERROR

WRITE DATA DO X' ' B' '
READ DATA DO X' ' B' '
READ DATA DI X' ' B' '
ERROR X (bad data bit)

ENTER TO CONTINUE
X TO RESTART
D TO GO TO DIAGNOSTIC LOOP
```

If no errors occur, card is good and following statement appears;

```
test complete
x to end
```

type x to get to main menu or
hit enter to get to main menu

Note: When test is complete the old card can be replaced with a new card and the test run on the new card. Each card should be tested for a minimum of 50 loops.

Diagnostic Mode

Used to determine source of error found in Stop on Error Mode.

There are three ways to get to this mode: From the main menu where you will enter the diagnostic data, from the Stop on Error Mode where the diagnostic data is that when the error occurred, or from the Single Step mode.

If you enter this mode from the Stop on Error or Single Step mode the diagnostic data which caused the error will be displayed along with a flag showing which bit or bits caused the error. This data will be continuously looped on untill Attn(>) EL is entered.

If you enter this mode from the main menu it will prompt you for the diagnostic data. Type in 4 hex characters of data you want to loop on.

This data will be continuously written and read back untill Attn EL is entered.

IBM Internal Use Only

Note: Screen for the diagnostic mode.

		Current menu/option	
		Current slot	
MODE = DIAGNOSTIC			
WRITE DATA DO	X'	'	B'
READ DATA DO	X'	'	B'
READ DATA DI	X'	'	B'
ERROR		X (bad data bit)	
ATTN EL TO END			

Test Cable

32 BIT DI/D0-A

DO BIT 15	(1)	————>	DI BIT 15	(30)
DO BIT 14	(20)	————>	DI BIT 14	(12)
DO BIT 13	(2)	————>	DI BIT 13	(31)
DO BIT 12	(21)	————>	DI BIT 12	(13)
DO BIT 11	(3)	————>	DI BIT 11	(32)
DO BIT 10	(22)	————>	DI BIT 10	(14)
DO BIT 9	(4)	————>	DI BIT 9	(33)
DO BIT 8	(23)	————>	DI BIT 8	(15)
DO BIT 7	(5)	————>	DI BIT 7	(34)
DO BIT 6	(24)	————>	DI BIT 6	(16)
DO BIT 5	(6)	————>	DI BIT 5	(35)
DO BIT 4	(25)	————>	DI BIT 4	(17)
DO BIT 3	(7)	————>	DI BIT 3	(36)
DO BIT 2	(26)	————>	DI BIT 2	(18)
DO BIT 1	(8)	————>	DI BIT 1	(37)
DO BIT 0	(27)	————>	DI BIT 0	(19)

Connect Digital Outputs to Digital Inputs On The Same Connector

The Number in parenthesis is the pin number to connect to.
ie. DI032-A pin (27) is connected to DI032-A pin (19)

B/M for building DI032-A Test Cable
Amp P/N Description

205210-1 (X1)	37 pin male D-Shell for mate to ADC-A
66506-9 (X37)	male pin
205731-1 (X1)	strain relief for 37 pin D-Shell
	Size 22 stranded Cable

Note: Use Crimper AMP PN 90302-1 or 90312-2 (RED DOT).

Figure 20. 32 Bit DI/D0-A Test Cable: Need only a 37 pin D-Shell
for the DI/D0 card. Male type

IBM Internal Use Only

Debug Procedure

- Place the DID032-A card to be tested into an open slot of the Ariel I/O Unit.
- Place the test cable on the DID032-A.
- Select IO(digital I/O) Test on the system main menu.
- Follow the flowchart in Figure 21 on page 39.

IBM Internal Use Only

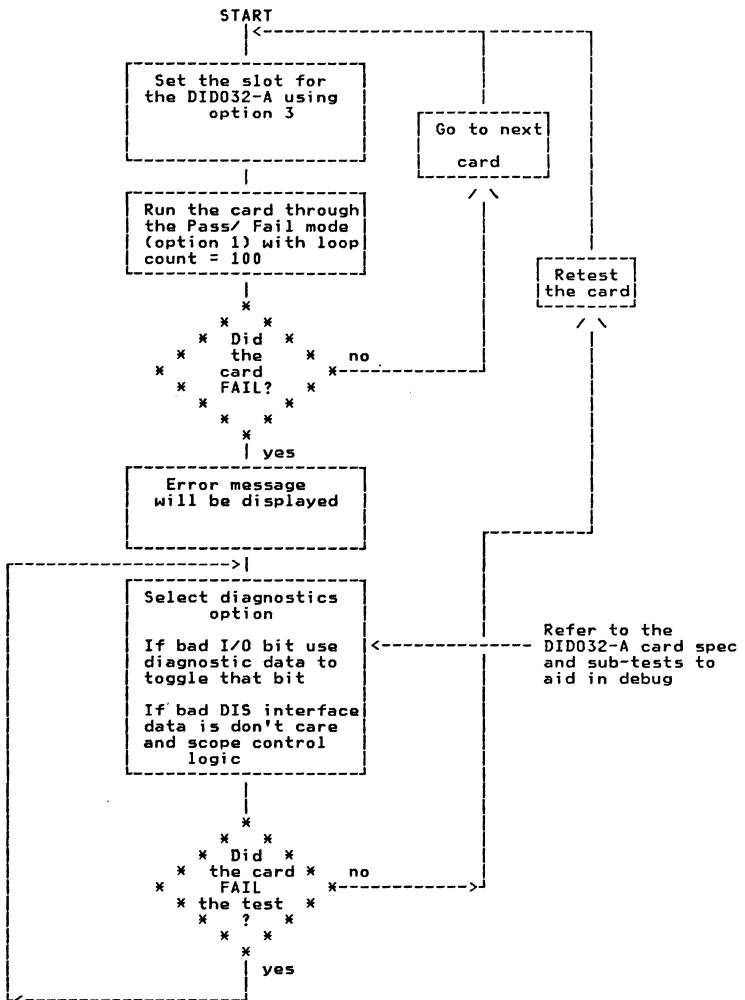


Figure 21. DID032-A Test Procedure

IBM Internal Use Only

(DC) DESTINATION COMMUNICATIONS ADAPTER CARD TEST

DCATA (Destination Communications Adapter Test software for Ariel) will be used for testing the functionality of the DCA-A card (PN 6912784). This program is written in EDX/J to run on a DSC IV. This program is loaded into the DSC IV when option "DC" is selected from the main menu of ATST. See Figure 9 on page 13.

Test Set-up

The set-up to test DCA-A is shown in Figure 22.

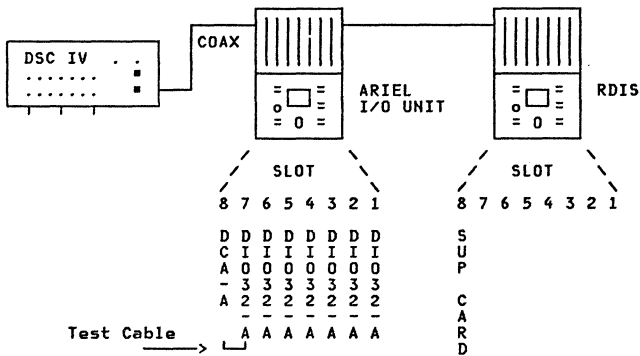


Figure 22. DCA-A Test Configuration

Block Address Test

There are eight switches on the card with the following function: Switches 1 through 5 in combination, determine the block address, with each switch having the following binary weight:

Sw 1 = 16
Sw 2 = 8
Sw 3 = 4
Sw 4 = 2
Sw 5 = 1

Switch 6 must be set to the off state for the DCA card to function. Switches 7 and 8 control the timing of the Operation Monitor.

Procedure:

- Set switches 1 through 8 to the off position
- Return to main menu of ATST
- Select CL for card list
- Hit ENTER on keyboard

IBM Internal Use Only

The display will list the I.D. of all cards in Block 0.

If, for example, there were only the DCA card in the unit, the display would show:

Block Macro 0000 ID=0001 BIC

This indicates that the card resident at block address 0, in macrofunction location 0 has an I.D. of 1, which is the Block Interface Card function, (implemented on the DCA card).

Set switch 5 on, for Block address 1, and hit ENTER.

The display will list the I.D. of all cards in Block 1.

Repeat the above, changing the block address by one and reading the corresponding display (in hexadecimal notation) until all addresses from 0 to 31 have been checked.

Test Program

Return to the main menu of ATST and select option DC to start the DCA card test program. This program requires a DI/DO card in each of the remaining card slots, and a test cable to interconnect one of the the DI/DO cards to the DCA card. The program utilizes two digital outputs and five digital inputs from the DI/DO card to interface to the DCA card under test. They function as follows:

D/O bit 0 generates an interrupt at Logical Space Address (LSA) 0

D/O bit 1 generates an interrupt at LSA 1 through 7, depending on which DI/DO card has the test connector attached

D/I bit 2 reads power sequence drive bit (BIC control reg. bit 1)

D/I bit 3 reads isolation drive bit (BIC control reg. bit 5)

D/I bit 4 reads enable op-mon bit (BIC control reg. bit 7)

D/I bit 8 reads operation monitor relay N/O contact

D/I bit 9 reads operation monitor relay N/C contact

After the test has been run successfully, the test cable must be connected to the next DI/DO card and the test repeated. The test is complete when each DI/DO card has been connected and exercised.

When the DCA Test is selected, the screen will display a message and prompt the user for input.

```
*****
*               *
*      IBM      *
*               *
*      DCA CARD TEST      *
*               *
*****
```

```
BLOCK=n
DO YOU WISH TO CHANGE BLOCK ADDR? __
```

Figure 23. Start of DCA card test

See Figure 23 for an example of the messages displayed. The block address from the previous test will be displayed, (shown in the figure as "n"). If the address of the block is correct then type N for no, proceed with the tests as directed by the program. If the block address is incorrect, type Y for yes, and the program will respond with the prompt to enter a block address within the range of 0 to 31. Enter the desired block address and proceed with the test.

IBM Internal Use Only

The corresponding block address must be set on the hardware by means of the switches 1 through 5 on the DCA card. Switch 6 must be set to the off state for the DCA card to function. Switches 7 and 8 control the timing of the Operation Monitor, and must be set as directed by the program during the Operation Monitor tests.

After the address has been set, the program prompts the user to hit ENTER when ready or EN to exit the test, and return to the main menu. To continue the test hit ENTER and the program will display the I.D. of each of the cards installed in the unit, starting with the DCA card in macrofunction location 0 with an I.D. of 0001.

The user will then be prompted to enter the address of the DI macrofunction card as follows:

- DI MACRO= n
- DO YOU WISH TO CHANGE MACRO ADDR? ____

Enter N if the number n corresponds to the DI/DO card connected to the DCA card via the test cable.

Enter Y if a different card address is used. Then enter the address of the DI/DO card to which the test cable is connected. The program will then perform the Control Tests.

Control Tests

The control tests write data to the BIC Control Register which resides in the Universal Block Interface module (UBI) on the DCA card. The state of the three user bits, bits 1, 5, and 7 are read as digital inputs to the DI/DO card via the test cable from the 9 pin D-shell connector on the DCA card to the input connector on the DI/DO card.

At the conclusion of the test, the display will be as follows:

- CONTROL TESTS
- TESTING POWER SEQ. DRIVE--2040 OK
- TESTING ISOLATION DRIVE--1040 OK
- TESTING OP-MON ENABLE-----0840 OK

The hex characters displayed on each line of the message above represent the data as seen by the DI/DO card as it reads the status of the three bits being tested. A different bit pattern will result in an ERROR message, indicating a failure of the associated signal.

The program will now advance to the Operation Monitor test.

Operation Monitor Test

The Operation monitor circuit provides the user with a set of relay contacts with which he may interface to alarm circuits in the tool. When the operation monitor is enabled, it activates the relay after a time delay. The user has a choice of four different delay periods which are set by switches 7 and 8 on the DCA card. The Operation Monitor Test prompts the user to set the switches to each of the four combinations, and then checks the state of the relay contacts by reading the DI/DO card bits to which the test cable is connected.

IBM Internal Use Only

The test is displayed as follows:

- TESTING OP-MON TIMING
- 0.105 SEC TEST SET SWITCH 7 OFF SWITCH 8 OFF
- HIT ENTER WHEN READY

When ENTER is hit, the following is displayed:

- TESTING INITIAL STATUS-----0040 OK
- TESTING TIME OUT TOO SOON--0840 OK
- TESTING TIME OUT-----0880 OK

The actual time out will then be displayed in seconds.

A failure will result in an ERROR message.

The test continues, prompting the user to set the switches for the other combinations, and checks for the proper timing interval. At the conclusion of the test, the program prompts the user to hit ENTER when ready, or EN to end test. ENTER will advance the program to the Interrupt test.

INTERRUPT TEST

The Interrupt Test writes the D0 bit 0 on, and reads the BIC to verify that the system interrupt to the DCA is functional. The test writes D0 bit 1 on, and verifies that the the interrupt from that particular card is functional. Each macrofunction card location must be tested to verify that all interrupts are functioning.

The display is as follows:

TESTING INTERRUPTS

- HIT ENTER WHEN READY EN to EXIT

After ENTER is hit the display is:

		READ	EXPECTED	
•				
•	TEST EXTERNAL INTERRUPT 0	0080	0080	OK
•	TEST EXTERNAL INTERRUPT 0	0080	0080	OK
•	TEST INTERRUPT FROM SLOT n	00XX	00XX	OK
•	TEST INTERRUPT FROM SLOT n	00XX	00XX	OK

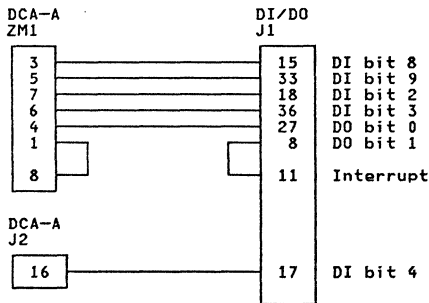
The bit pattern represented by XX above depends on the location of the DI/D0 card in the test, with (n) the number of the slot where the card resides. An ERROR message is displayed if the bit pattern read does not match the expected pattern. The program provides the option to loop on the interrupt test, and displays the number of loops and the number of errors.

The Interrupt test concludes the DCA card test.

- type EN and hit ENTER to exit back to main menu
- disconnect the 37 pin test connector from the DI/D0 card
- connect the 37 pin test connector to the next DI/D0 card
- select DC from menu and hit ENTER
- perform the DCA card test again

IBM Internal Use Only

Repeat the DCA card test with a DI/D0 card for each card location in the unit, to verify that all address combinations and interrupts are functional.



B/M for building DCA A Test Cable with size 22 stranded wire

Amp P/N	Qty	Description
205210-1	1	37 pin d-shell connector
205731-1	1	strain relief
205204-1	1	9 pin d-shell connector
205729-1	1	strain relief
66506-9	15	male pin
2-530831-7	1	30 pin card connector

Figure 24. DCA-A Test Cable: The 37 pin connector mates with J1 on the DI/D0 card, the 9-pin connector mates with ZM1 on the DCA-A card and the 30 pin connector mates with J2 on top of the DCA-A card

See Figure 24 for an illustration of the cable required to run the DCA-A Card Test.

Repeater Function

Data received on the input co-axial cable, is amplified and re-transmitted over the output co-axial cable to the next unit. To test this repeater function, an RDIS unit is connected to the output co-axial connector of the Ariel under test, and a message is sent from the DSC. A proper response from the RDIS indicates the the DCA card repeater function is working.

Repeater Test Connections

- Co-axial cable from DSC to bottom co-axial connector on DCA card
- Co-axial cable from top co-axial connector on DCA card to bottom connector of RDIS

Repeater Test

Return to the main menu of ATST and select CL for card list. This will list all of the cards in the Ariel unit, and, if the repeater function is working, all of the cards in the RDIS as well. If the program fails to list the RDIS cards, change the address of the Ariel via switches 1

IBM Internal Use Only

through 5 on the DCA card. If the Program still fails to list the RDIS cards, perform the following checks:

- Ariel power is on
- RDIS power is on
- Co-axial cables connected correctly

Continued failure indicates a bad component on the DCA card. Refer to the Diagnostic section for problem resolution.

Bypass

The input and output co-axial cables are connected to circuitry on the card by means of relay contacts. In the event of a failure of one of the units in a system, the malfunctioning unit may be bypassed, switching the signal from the input coaxial connector directly to the output coaxial connector. This is done by connecting pin 2 to pin 9 on the 9 pin d-shell connector.

Bypass Test

To test the bypass function, proceed as follows:

- Turn Ariel power switch off
- Disconnect the coaxial cables at the unit
- Remove cable at 9 pin d-shell connector

Perform the following tests using an ohmmeter or other device suitable for checking continuity:

- Verify that continuity exists between the center contacts of the two coaxial connectors on the card
- Verify that continuity exists between the outer shells of the two coaxial connectors on the card

Turn Ariel power switch on

- Check that continuity is lost between the center contacts of the two coaxial connectors on the card
- Check that continuity is lost between the outer shells of the two coaxial connectors on the card
- Connect pin 2 to pin 9 on the card mounted 9 pin d-shell connector
- Verify that continuity exists between the center contacts of the two coaxial connectors on the card
- Verify that continuity exists between the outer shells of the two coaxial connectors on the card

Diagnostics

Preliminary Tests

Should any of the tests fail, the inputs to the card should be checked before proceeding with the diagnostics.

IBM Internal Use Only

Input Voltages

Verify that the D.C. voltages to the card are present at the following points:

- + 8.5 volts at bottom connector J1 pin 33
- - 5 volts at bottom connector J1 pin 71
- + 5 volts at top card connector JJ2 pin 1
- + 1.7 volts at top card connector JJ2 pin 4

The + 1.7 volts is generated on the card and should be 1.7 plus or minus five percent. If this voltage is not present, check the following:

- Regulator module U2 pin 3 should be approximately + 3 volts
- Transistor QT1 emitter should be approximately + 2.5 volts

If these readings are correct, the probable cause of failure is transistor QT2 and QT3. If the voltages are not correct, replace QT1.

Clock Oscillator

Using an oscilloscope:

- Set time base to 0.05 microseconds per Cm.
- Verify 20 Mhz oscillator output present at top card connector JJ2 pin 13.
- Replace oscillator U1 if there is no output.

Data Input

Verify that data is being received by the card as follows:

- Set all switches on the DCA card to OFF.
- Set switch 5 on, for a block address= 1.
- Return to the test menu of ATST.
- Select CL for card list.
- Select option L to loop on this block.
- Set oscilloscope time base to 10 microseconds per Cm.

Monitor top card connector JJ2 pin 9 with the oscilloscope probe, and observe that a burst of serial data is present. This is the output of the receiver amplifier, U12, and indicates that the analog input circuit is functioning. If there is no output at this pin, monitor the co-ax input at connector ZM2, and at top connector JJ2 pin 19. If data is observed at ZM2 but not at JJ2-19, it indicates relay 2 is not closed. If data is present at both ZM2 and JJ2-19, but not at JJ2-9, it indicates a malfunction of amplifier U12 or it's associated circuitry.

Debug

Symptom: Card fails to return it's I.D when addressed.

Check following:

IBM Internal Use Only

- Input data at JJ2 pin 9.
- Switches 1 through 5 for proper block address.
- Switch 6 off.
- Relays 2 and 3 are energized.

Symptom: Bypass Test fails.

Check following:

- Relays 2 and 3.
- Transistors QT10, QT8, QT9.

Symptom: Repeater Test fails.

Check Following:

- Relays 2 and 3 are energized.
- Data present at collector of transistors QT11, QT12, QT13.

Symptom: Control Test portion of DCA test fails.

- Remove test cable at 9-pin d-shell.
- Set DCA card switch 4 on, all others off, for block address 2.
- Return to main menu of ATST program.
- Select option MF for macrofunction test.
- Choose option 3 to write data to card.
- Enter block and macro 020.
- Enter CTC + count B1.
- Enter data FF.
- Monitor pin 6 of ZM1 (DCA card 9-pin connector) with oscilloscope.
- Pin 6 should be at a low logic level.
- Repeat option 3 above with data equal to 00.
- Pin 6 should go to a high logic level.

Failure of pin 6 to respond to changes in data indicates malfunction in module U14 or associated wiring.

Symptom: Operation Monitor fails.

- Check that switches are set as per program instructions.
- Check test cable between DCA and DI/D0 cards.
- Monitor JJ2 pin 17 with oscilloscope.

JJ2 pin 17 output should be a square wave of approx. 210 Ms. when switches 7 and 8 are off. Changing the switch settings should result in an increase in the duration of the square wave. No output indicates a malfunction in the timing circuit, comprised of modules U6, U7, U8, and U9.

IBM Internal Use Only

{P0} PULSED OUTPUT MODULATION CARD TEST (TPAC-A)

POMTA (PCM Test Software for Ariel) will be used for testing the functionality of the TPAC-A card (PN 6912778) with POM software as its application. This program is written in EDX/J to run on a DSC IV. This program is loaded into the DSC IV when option "P0" is selected from the main menu of ATST. See Figure 9 on page 13.

```
*****
*                               *
*               I B M           *
*               POM CARD TEST   *
*               time      date   *
*                               *
*****
----- MAIN MENU -----
LSA = 0000 CARD ID = 0033
1 READ / WRITE TEST
2 SET DUTY CYCLES
3 ENABLE / DISABLE OUTPUT
4 DISPLAY OUTPUT STATUS
5 DISPLAY DUTY CYCLES
6 DISPLAY OUTPUT LINE STATUS
7 STATUS CONTROL
9 CHANGE LSA
99 END TEST                ** ENTER OPTION ==> __
```

Test Procedure

The functionality of the TPAC-A is for general purpose use. Thus when an application is needed, software is generated specifically for that application. And since the card is a processor the application code contains bring-up diagnostics. This code not user dependent, tests the functionality of the card except for analog circuitry and the user ports. Thus, the test procedure for this card will be to test the user ports for the application it is being assembled as.

This test generates POM outputs and then reads back what was generated by the status lines. The power percentage (on time) is varied through the test, starting at 1% and going to 99%. Any failures will be printed to the 3101 console. For more information on POM see PN 6290866 Form No. E45-1381-1 -- Pulsed Output Modulation Macrofunction Card Functional Specification.

Note: There are only 8 lines available. The code was written for up to 16 but at this time hardware does not allow it.

The following is the test procedure for TPAC-A with POM code.

1. Option Description
2. 9 Select the correct LSA
3. 1 The read/write test is a Pass/Fail test
 Select 8 lines (16 - NA)
 Select L -- Takes 5 minutes
4. If card passes continue with next card
5. If card fails use other options to test

Test Cable

IBM Internal Use Only

DSCR	PIN	TPAC-A	DSCR	PIN
DI BIT 0	(1)	—————>	DO BIT 0	(22)
DI BIT 1	(14)	—————>	DO BIT 1	(10)
DI BIT 2	(2)	—————>	DO BIT 2	(23)
DI BIT 3	(15)	—————>	DO BIT 3	(11)
DI BIT 4	(3)	—————>	DO BIT 4	(24)
DI BIT 5	(16)	—————>	DO BIT 5	(12)
DI BIT 6	(4)	—————>	DO BIT 6	(25)
DI BIT 7	(17)	—————>	DO BIT 7	(13)

**Connect Digital Outputs to Digital Inputs
On The Same Connector**

The Number in parenthesis is the pin number to connect to.
ie. TPAC-A pin (17) is connected to TPAC-A pin (13)

B/M for building TPAC-A Test Cable
Amp P/N Description

205208-1 (X1)	25 pin male D-Shell for mate to TPAC-A
66506-9 (X25)	male pin
205718-1 (X1)	strain relief for 25 pin D-Shell Size 22 stranded Cable

Note: Use Crimper AMP PN 90302-1 or 90312-2 (RED DOT).

Note: The 9 pin D-Shell is for communications and will be tested using a ROC.

Figure 25. TPAC-A Test Cable For POM CODE: Need only a 25 pin D-Shell for the TPAC-A card (Male type).

(RC) ROC CARD TEST (TPAC-A)

ROCTA (ROC Test Software for &ar) will be used for testing the functionality of the TPAC-A card (PN 6912778) with Auto-poller software for communications to the ROC as its application. This program is written in EDX/J to run on a DSC IV. This program is loaded into the DSC IV when option "RC" is selected from the main menu of ATST. See Figure 9 on page 13.

```

*****
*
*          AUTO POLLER/ROC DIAGNOSTIC OPTION MENU          *
*
***** REL 2.0 - 04/04/86-08.37 *****
1- POLL LIST GENERATION          7- TABLE LIST GENERATION
2- XMIT DATA                    8- WRITE/READ LOOP
3- READ DATA                    9- READ STATUS
4- POLL MODE                     10- LIGHT TEST
5- CHANGE ADDRESS                11- END
6- AUTO TEST

```

ENTER OPTION #:

See the AUTO POLLER FUNCTIONAL SPECIFICATION for option descriptions.

IBM Internal Use Only

Test procedure

1. Select option 5

Set LSA (block and macro) for the correct address of TPAC-A

2. Select option 1

2 questions will be asked

- a. Enter # of terminals on poll list: ---> 1
- b. Enter terminal address: ---> 11

3. Select option 7

This will return with 'TABLE LIST GENERATED'.

4. Select option 8

This does a read write test to the ROC terminal. The ROC display should be flashing letters and numbers.

(MF) MFTJ

This test allows the user to do single DIS reads or writes to a selected card. Once the card is selected, it will stay selected until another card is selected via CTC F. This has been edited from the original MFTJ to be compatible with RDIS and the Ariel I/O unit. It is not recommended to run on a DIS system.

The Attn L0 command must be done before the following is displayed on the screen:

```
"ENTER CTC + COUNT (HEX CC _ 0000) = "
```

If it isn't then the loop will occur on the next selection (i.e. 1, 2, or 3 of the MFTJ main menu). Thus to loop on a data pattern for either a read or a write the ATTN L0 cmd should be entered directly after selecting an option from the MFTJ main menu.

(BU) BURN IN TEST

This option is unlike the others because it resides in ATST. It is another test program but is loaded with ATST. This program runs on the DSC IV when option "BU" is selected from the main menu of ATST. It should be noted that Card List (option CL) must be run before running burnin. See Figure 9 on page 13.

After running Card List on the desired blocks, a map of the Macro-Function Cards on the DIS channel is generated. Once the map is created, BU will attempt to read the ID of every possible LSA (256) on the DIS channel. An error will be recorded every time a return is generated from a LSA select that is not in the map generated from Card List, or every time a return is not generated from a LSA select this is in the map. For example;

- 1. If a card/block is added to the system after Card List had been run, that card will generate errors.
- 2. If a card/block is taken out of the system after Card List had been run, then errors will be generated and logged.

BU function is to select the LSA and read the ID. There are no data patterns used to write, read and then compare.

IBM Internal Use Only

(CL) CARD LIST

This option is unlike the others because it resides in ATST. It is another test program but is loaded with ATST. This program runs on the DSC IV when option "CL" is selected from the main menu of ATST. See Figure 9 on page 13.

This program is used to locate what blocks and Macro-Function Cards exist on the DIS channel. When selected, a description of the next block will be displayed on the 3101 screen. An example is described in Figure 26 on page 52.

(RW) READ WRITE TEST

RWTA (Read Write Test for Ariel Burn in) will be used for burning in the I/O units. This Test program is an enhanced version of the BU (BUrnin) option and is written in EDX/J to run on a DSC IV. This program is loaded into the DSC IV when option "RW" is selected from the main menu of ATST. See Figure 9 on page 13.

This test program was designed to exercise the DIS channel as heavily as possible. For each card connected to the DIS channel, remote and local, it selects the card, writes a test pattern to 2 of the card's registers, reads back the data from the same registers, compares the data with that written, stores the error code in a table, and then selects the next card in the map and tests it, etc.. To function properly, a map of the I/O cards/blocks on the DIS channel must first be generated. RWTA automatically selects the first block on the DIS channel and displays the Macro-Function Cards placed in that block as shown in Figure 26 on page 52. After hitting enter on the 3101 keyboard, the next valid block on the DIS channel will be displayed with it's Macro-Function Cards (e.g. If there are only 3 blocks of the 32 possible connected on the DIS channel, only 3 screens will be displayed. The user does not have to go through all 32 blocks like Card List CL option). After all blocks have been searched, RWTA test begins. If errors occur they will be displayed on the 3101 screen as shown in Figure 27 on page 53 and logged into the error log in Hazel-A memory. Only those cards/blocks found when creating the map are tested. Thus errors will only be logged if a return is not generated for an LSA that exists in the map. Also, all errors are time stamped to the nearest minute.

The test patterns generated for the 2 registers for each card connected to the DIS channel are listed below.

1. Write X'FF' to the registers then shift left 1 bit at a time while padding with a B'0'
2. Write X'00' to the registers then shift left 1 bit at a time while padding with a B'1'
3. Write X'FF' to the registers then shifts left a B'0' through, 1 bit at a time
4. Write X'00' to the registers then shifts left a B'1' through, 1 bit at a time

After the test pattern is generated, each card is tested and then a new pattern is generated. Thus each card gets the same pattern before the pattern is changed.

When ending RWTA (attn EL), the error log will be displayed on the 3101 screen as shown in Figure 28 on page 54. As shown, there are 2 different error types. They are:

DIS ERRORS These errors represent DIS errors while accessing the Card.

DATA ERRORS These errors represent data errors from the card. This is not a DIS error, but rather a bad compare of the data read with that written.

IBM Internal Use Only

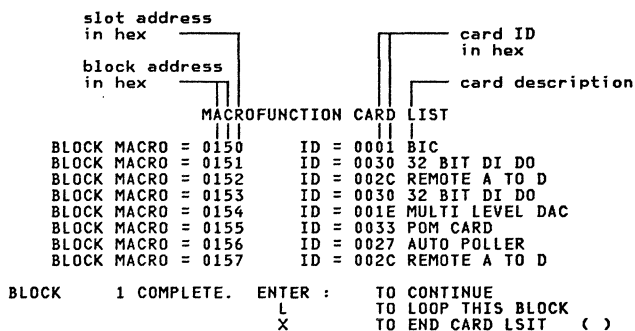


Figure 26. Card List Screen: This is only an example of what might exist in one of the blocks on the DIS channel. Only those cards that are placed in the block will be displayed. For example, if a card does not exist, then only 7 Macro-Function Cards will be displayed.

IBM Internal Use Only

```

ERR COUNT   LSA      CTC+ID    WRITE    READ
READ WRITE ERROR    1     0025    0F30    0000    0000
READ WRITE ERROR    2     0026    0F2C    0000    0000
READ WRITE ERROR    3     0027    0F33    0000    0000
READ WRITE ERROR    4     0021    0F30    0000    0000
READ WRITE ERROR    5     0022    0F30    0000    0000
READ WRITE ERROR    6     0023    0F30    0000    0000

Error count  <-----|-----> Data read
max = 64,000         |-----> data error only
                     |-----> data written
Block Address        <-----|-----> data error only
Macro Address        <-----|-----> Card ID
                     |-----> CTC when failure occurred
```

Figure 27. ERROR DISPLAY While RWTA is running

IBM Internal Use Only

DIS errors while reading the card ID → BURN LOOP COUNT = 49278 < — # of loops for each test pattern
 LSA ERRORS / CARD
 BLOCK MACRO = 0020 7 ERRORS
 BLOCK MACRO = 0021 7 ERRORS
 BLOCK MACRO = 0022 7 ERRORS
 BLOCK MACRO = 0023 7 ERRORS
 BLOCK MACRO = 0024 6 ERRORS
 BLOCK MACRO = 0025 6 ERRORS
 BLOCK MACRO = 0026 7 ERRORS
 BLOCK MACRO = 0027 7 ERRORS
 HIT ENTER
 This shows the total number of errors for each card. DIS, and Data errors

when 'ENTER' is hit the next report is shown

READ WRITE ERROR REPORT
 DIS ERROR COUNT = 0, DATA ERROR COUNT = 6
 HIT ENTER
 Number of DIS Errors encountered while running diagnostics
 Number of Data Errors encountered while running diagnostics

when 'ENTER' is hit the next report is shown

ERR	LSA	CTC+ID	WR	RD	MO/DAY	HR:MN
1	0024	002C	0060	007F	4/27	2:30
2	0012	0030	FF80	FFFF	4/27	2:30
3	0013	0033	FF80	FFFF	4/27	2:30
4	0016	002E	FF80	FFFF	4/27	2:30
5	0017	0030	FF80	FFFF	4/27	2:31

Only the last 500 errors are saved for viewing

150 0021 0030 FF80 FFFF 4/27 2:31
 OPTIONS : ENTER , ERR # , OR 999 TO END < — See notes 1,2,3 below
 Error number in error log max=500
 Block and Macro of card with error
 The first byte is the ctd that the error occurred on, if the error is a DIS error
 The second byte is the ID of the card where the error occurred
 Date Time
 Diagnostic data read back from the card
 Diagnostic data written to the card

- 1 - Hitting enter will display the next set of errors.
- 2 - Typing an error number will take you to that error number.
- 3 - Entering a number larger than that recorded or 999 will end this menu.

Figure 28. RWTa Error Log

After browsing the error log the RWTa option menu is displayed as shown Figure 29 on page 55. The options are described below;

IBM Internal Use Only

END READ WRITE

R TO RESHOW ERRORS
D TO DISP. ERR/LSA
S TO START TEST NEW
C TO CONTINUE TEST
X TO END CARD LIST

ENTER OPTION (

Figure 29. RWTA Option Menu:

Option	Description
R	Re-show error log (RWTA test not running)
D	Will show only the errors from the error log for the LSA entered. When enter is hit after selecting this option the user will be prompted for an LSA.
S	Reset error log, and go to beginning of RWTA (card list).
C	Continue testing with same DIS channel map (do not reset error log).
X	End RWTA altogether.

(PL) PLANAR BOARD & SUPPORT CARD TEST

(RDIS ONLY) This section refers only to the R-DIS

This section will describe the test software for the Remote DIS (R-DIS) Planar bd and Support Card. It includes information regarding the hardware required to do test and debug, the tests which will be performed by the test program and a debug chart which can be used to aid in fault isolation. This specification provides information for the programmer regarding the software structure and the interaction with the test operator/technician.

It is assumed that this program is a part of a larger program and that this program will be written in the EDXJ language and will be run on a &dsc, or ARIEL Controller. Also, the user must be knowledgeable of the Remote DIS and the present DIS system.

General Information

Required Hardware:

- IBM Test System
- Working IBM Power System pn.6227050
- Good Support Card pn.6227047
- Good DI/DO Card pn.6227041
- Coax Cable Asm. RG-62 au
- Digital Multimeter
- Oscilloscope

Required Documentation

- Remote DIS Functional Specification
- Planar Board Schematic pn.6227802
- Planar Board Assembly Dwg. pn.6308497
- Planar Board Card Detail pn.6227801

IBM Internal Use Only

Software

2 functions are required during initialization:

1. Initialize the BIC
2. Disable the Watchdog Timer

All test routines are to provide the operator with optional loop on error.

All DIS errors are to be displayed on screen, informing the operator that there is a test system hardware problem.

Main Menu

1. +1.7 volt test / 20 mhz oscillator test
2. Parallel I/O Test
3. Serial I/O Test
4. Watchdog Timer Test
5. Macro Test
6. Power Sequence Drive Test
7. End Planar Board test

The test program will be selected via the Main Test Menu, by selecting Planar Board (P/N 6308497) Test. Once the test is loaded, a main menu listing all the Planar Board tests will be displayed.

The test operator is to select all tests sequentially beginning with test 1. The operator is to follow all prompts and instructions appearing on the the screen.

The Planar Board is said to be 'good' if all tests have been run successfully. The board is 'bad' if 1 or more tests fail. The operator is required to run all tests even if a previous test has failed. A data sheet, listing the failed tests is to be put with the board and both are then to be set aside for later debug.

The main menu provides the means for running 6 separate tests on the board, a 7th option provides a program exit. The tests are described as follows.

Note: All tests have a loop on error feature. If any test fails, see "Debug Chart" on page 59.

1. +1.7 volt test / oscillator test

This test is performed to verify 2 functions:

- a. The on-board temperature sensitive voltage regulator (Z6) is regulating +1.7 volts.

"Connect Digital Voltmeter to Connector J12 as follows:"
Positive Lead to pin 33

Negative Lead to pin 5

"If the voltage reading is +1.7 vdc, the test is passed."

"hit enter to continue"

If enter is pressed the following is displayed.

"Oscillator Test"

- b. The oscillator is providing a 20 MHZ square wave.

"Connect Scope Probe to J12-28 and verify a 20mhz square wave"

"Hit 'enter' to return to main menu"

IBM Internal Use Only

2. Parallel I/O Test

This is not implemented

3. Serial I/O Test

This test verifies that all serial data paths from/to J8 and J9 to/from Z5 are functional. The serial internal circuits of Z5 are also verified by this test.

The following text is required.

"Connect serial i/o coax cable to 'in' on the support card"

"Set DIP Switch 6 to 'off'

"Hit 'enter' to continue"

If 'enter' is pressed the following is required:

- Invoke standard BIC diagnostics using the chart on page 17 of the Remote DIS Spec.
- Exercise each CTC sequence with standard test patterns eg. X'aa", x'55', x'01', x'02', x'04' etc.
- Display whether test has passed or failed and then the following

"Hit 'enter' to return to main menu"

4. Watchdog Timer Test

This test verifies that the watchdog timer modules Z7, Z8, Z9, Z10, Z11 and Z12 are functional. Also verified by this test is the opmon signal to J9 to the support card.

Assume that the watchdog timer has been disabled by the init code.

The following text is required

"Connect an ohm meter to the 25 pin 'd' shell of the support card as follows:"

"relay common --- pin 12"

"relay normally open --- pin 10"

"Verify that these points are 'open'"

"hit 'enter' to continue"

"Disconnect ohm meter lead from pin 10 and connect to pin 11"

"Verify that these points are 'closed'".

"hit 'enter' to continue"

"Set DIP switches 7 & 8 to the 'off' position which will select the .157 second time out"

The following routines are to be invoked:

- Enable watchdog timer
- Attach watchdog reset task

Display the following:

"Hit any key to start test, contacts will 'make' and/or 'break' after the selected time out time."

"Verify that the relay contacts have switched"

IBM Internal Use Only

All of the above is to be repeated for the remaining 3 timeouts which are as follows:

SW-7	SW-8		
off	off	---	.105 sec
off	on	---	.420 sec
on	off	---	1.680 sec
on	on	---	3.350 sec

5. Macro Test

The test verifies that all data paths to/from J1 thru J7 from/to Z5 are functional.

The purpose of this test is to verify that slots 1-7 are functional. To accomplish this a known good DI/D0 card is required. The card is to have it's wrap test cable inserted into the 37 pin 'd' shell. The card is then to be plugged into slot 1 and the DI/D0 diagnostic is to run. The above is to be repeated for slots 2-7.

All appropriate text messages prompting the operator are to be put on the screen.

6. Power Sequence Drive Test

This test verifies that this signal is functional from Z5 to the support card through J8.

Display the following text.

"Power Sequence Drive Test"

"Connect scope probe to the support card's 25 pin 'd' shell pin 9."

"Hit 'h' to bring the line 'hi'"

"Hit 'l' to bring the line 'lo'"

"Verify that this line is functional"

"Hit 'enter' to continue"

IBM Internal Use Only

Debug Chart

Symptom	Possible Cause	Procedure
+1.7 Volt Failure	open connection to support card	see schematic sh 2 of 5 check all paths to support card.
	no power to Z6	check power to Z6
	no support card	insert support card
	wrong Z6 orientation	Check orientation and PN Re-orientate or replace Check capacitor orient.
Oscillator Failure	no power	check power
	bad oscillator	replace
Serial I/O Failure	Problem in serial data path	Verify serial data at J12-30, 31, 32.
	Problem on common I/O bus caused by either Z5 or Z1, Z2, Z3, Z4	see schematic sh 1 of 5 probe bus
	DIP Switch 6 in 'on' position	Put DIP switch 6 into 'off' position
Watchdog Timer Failure	Watchdog Clear circuit problem	check circuit on schematic sh 1 of 5
	watchdog counter problem	check circuit on schematic sh 2 of 5
	multiplexer problem	sh 2 of 5
Macro Test Failure	cold solder connection	If problem with 1 connector, resolder
	bad circuit	probe bus on schematic sh 1 of 5
Power sequence dr. failure	no signal from Z5 to support card	check signal and signal path from Z5 Schematic sh 1 of 5

Use Schematic and Assembly Drawings for the locations of Zx and Jxx.

SUPPORT CARD

This card test falls under the Planar Bd test option (PL) on the main menu. The test procedure, function and debug of the card is described below.

- Connect known good Planar Board to IBM Power system (and power system to a suitable power source.)
- Have IBM supplied test system running nearby (set up to run 'Planar Board Test'.)

IBM Internal Use Only

Circuits to be tested

1. POR
2. +1.7 volt regulator
3. Serial I/O (Communications Adapter)
4. Watchdog Timer
5. Power Sequence Drive signals

Procedure for required tests

The following is a list of steps to be taken to test a RDIS Support Card.

1. Set the DMM to read OHMS (use lowest scale---200 ohm on most DMM's).
2. Measure continuity from J3 (pin 20) to J2 (pin D08). (J3 is the gold "fingered" connector along the long edge of the card, J2 is the connector in the green housing along this edge).
3. Measure continuity from J4 (inner conductor) to J5 (inner conductor). (J4 and J5 are the two coaxial connectors at the edge of the card. Continuity means a reading of less than 5 ohms).
4. Measure continuity from J4 (outer conductor) to J5 (outer conductor).
5. Plug Support Card into previously connected Planar Board.
6. Apply power to system (i.e. plug it in and turn it on).
7. After 3-4 seconds, measure NO continuity from J4 (outer conductor) to J5 (outer conductor).
8. Measure NO continuity from J4 (inner conductor) to J5 (inner conductor).
9. Remove power (turn RDIS unit off).
10. Measure continuity from J4 (inner conductor) to J5 (inner conductor).
11. Measure continuity from J4 (outer conductor) to J5 (outer conductor).
12. Set DMM to a range that will permit measurement of 20 VDC.
13. Apply power to system.
14. Connect DMM's positive lead to Support card's J3 (pin 20) and negative lead to Planar Board's J12 (pin 5) and measure 3.5 VDC or greater.
15. Connect DMM's positive lead to Planar Board's J12 (pin 33) and leave negative lead connected to Planar Board's J12 (pin 5) and measure 1.7 VDC. DMM may then be disconnected.
16. Run Planar Board Test #3.
17. Run Planar Board Test #4.
18. Run Planar Board Test #6.

IBM Internal Use Only

Debug Chart

Failed Test	Probable Cause	Other Possibilities
3,4	Bad relay(s)--K1,K2	Coax connector(s)--J4,J5
2	Bad relay--K3	
+1.7 Volt Failure	open connection to support card bad component or poor connection	see schematic check all paths to/from planar board. Check and/or replace: Q16,17,18, R20,21
Serial I/O Failure	Problem in serial data path DIP Switch 6 in 'on' position	Verify serial data at Z9,pins 4 & 9 Put DIP switch 6 into 'off' position
Watchdog Timer Failure	Opmon driver circuit problem	check circuit on schematic
Power sequence dr. failure	no signal from planar board	check signal and signal path from planar board

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

APPENDIX A. GLOSSARY

RDIS: Remote Distributed Interface System.

DIS: Distributed Interface System.

DSS: Distributed Sensor Subsystem -- Comprised of a DSC IV or Ariel and DIS.

ARIEL: Follow-on product of the DSS.

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

APPENDIX B. SOFTWARE LIBRARIES

DSC IV SOFTWARE LIBRARY

- Source Library --> 'TWB036.IUARIEL.ASM'
- Load Library --> 'TWB036.IUARIEL.LOADLIB'

Note: Any person considering writing a new member for this library should consult Art Scanlon, or Tad Burnett D/035.

SOURCE	LOADLIB	
ATST	GCODE	Main program
AITA	AIT4	Analog Input Test (ADC)
AOTA	AOT4	Analog Output Test (DAC)
DCATA	DCAT4	Destination Comm Adpt Test (DCA-A)
PLANA	PLAN4	Planar Board Test (Mother BD and Sup CD)
IOTA	IOT4	DI/DO Test (32 Bit DI/DO CD)
MFTJA	MFTJ5	Macro-Function Test Program
POMTA	POMT4	Pulsed Output Modulation Test (TPAC)
ROCTA	ROCT4	Remote Operator Console Test (TPAC)
RWTA	RWT4	Diagnostics for testing system
EPRM4A	EPRM4	Eprom Diagnostics (EPRM4-A)
BURN2	BURN4	Eprom Burn Program (EPRM4-A)

ARIEL SOFTWARE LIBRARY

- Source Library --> 'TWB036.IUARIEL.ASM'
- Load Library --> 'TWB036.IUARIEL.LOADLIB'

SOURCE	LOADLIB	
ATST	ATST	Main program
AITA	AITA	Analog Input Test (ADC)
AOTA	AOTA	Analog Output Test (DAC)
DCATA	DCATA	Destination Comm Adpt Test (DCA-A)
PLANA	PLANA	Planar Board Test (Mother BD and Sup CD)
IOTA	IOTA	DI/DO Test (32 Bit DI/DO CD)
MFTJA	MFTJA	Macro-Function Test Program
POMTA	POMTA	Pulsed Output Modulation Test (TPAC)
ROCTA	ROCTA	Remote Operator Console Test (TPAC)
RWTA	RWTA	Diagnostics for testing system
EPRM4A	EPRMA	Eprom Diagnostics (EPRM4-A)
BURN2	BURNA	Eprom Burn Program (EPRM4-A)

Software Conventions

1. The same source is used for the DSC IV and ARIEL Controller .
2. ATST is the only task running. All attention routines must be added to ATST , not the program being added.
3. DSC IV loadlib names end in a "4".
4. ARIEL Controller loadlib names end in an "A".

Heading ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Heading References</u>
intro	INTRO	1	INTRODUCTION
thard	HARDWARE	1	TEST HARDWARE SET UP
sucable	HARDWARE	2	System Setup Cables
tca	HARDWARE	4	Test Cables (All Cards)
sysover	OVERVIEW	4	Ariel SYSTEM OVERVIEW
convref	SOFTWARE	11	SOFTWARE GENERAL SPECIFICATION
atst	ATST	13	ATST
ai	ATST	13	(AI) ANALOG INPUT CARD TEST
aitc	ATST	24	Test Cable
ao	ATST	27	(AO) ANALOG OUTPUT CARD TEST
aomm	ATST	27	AOTA Main Menu
opr	ATST	28	Test Operation
util	ATST	29	DAC-A Utility Menu Options
aotc	ATST	31	Test Cable
io	ATST	34	(IO) DIGITAL INPUT/OUTPUT CARD TEST
iotc	ATST	37	Test Cable
dc	ATST	40	(DC) DESTINATION COMMUNICATIONS ADAPTER CARD
po	ATST	48	TEST (PO) PULSED OUTPUT MODULATION CARD TEST (TPAC-A)
rc	ATST	49	(RC) ROC CARD TEST (TPAC-A)
mf	ATST	50	(MF) MFTJ
bu	ATST	50	(BU) BURN IN TEST
cl	ATST	51	(CL) CARD LIST
rw	ATST	51	(RW) READ WRITE TEST
deb	PLANAR	59	Debug Chart
deb2	PLANAR	61	Debug Chart
lib1	MAIN	65	Appendix B. Software Libraries

Figure ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Figure References</u>
config2	HARDWARE	2	1: 4, 4
cableh	HARDWARE	3	2: 2
cabler	HARDWARE	3	3:
cablet	HARDWARE	3	4:
cabled	HARDWARE	4	5: 2
mbd	OVERVIEW	5	6: 4
pns	OVERVIEW	6	7: 4
dsciv	OVERVIEW	7	8: 6
mmenu	ATST	13	9: 14, 27, 34, 40, 48, 49, 50, 51, 51
aimenu	ATST	15	10:
aitfmat	ATST	16	11: 15
ai2menu	ATST	17	12: 15, 15
ai3menu	ATST	17	13: 15
ai4menu	ATST	17	14: 15
ada	ATST	24	15: 14
aitp	ATST	26	16: 25
aomenu	ATST	27	17:
ao4menu	ATST	29	17:
daa	ATST	32	17: 27
aotp	ATST	33	18: 32
iomenu	ATST	34	19: 34
dio	ATST	37	20:
iotp	ATST	39	21: 38
dcats	ATST	40	22: 40
testrt	ATST	41	23: 41
dcaa	ATST	44	24: 44
pomenu	ATST	48	25:
tpio	ATST	49	25:
rcmenu	ATST	49	26:

cldsp	ATST	52	26:	51, 51
scrn	ATST	53	27:	51
rwtaer	ATST	54	28:	51
rwtaom	ATST	55	29:	54

Imbed Trace

Page vi	INTRO
Page 1	HARDWARE
Page 4	OVERVIEW
Page 10	SOFTWARE
Page 13	ATST
Page 55	PLANAR
Page 63	GLOSSARY
Page 65	LIB

32 Bit DI/DO-A Hardware Functional Specification

Control Systems Engineering

**IBM Corporation
General Technology Division
D-035 B-300-47A
East Fishkill
tel 533-7654**

IBM Internal Use Only

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

TABLE OF CONTENTS

32 Bit DI/DO Card	1
General Description	1
Abstract	1
Features	1
Part Numbers and Related Documents	2
A-BUS Interface Definition	2
A-BUS Inputs and Outputs	2
Second Level Logics	3
Hardware	5
Timing	6
Command Tag Definitions	7
I/O	8
Hardware Program Jumpers	9
Programming the Digital Outputs	10
Miscellaneous Hardware Isolation Procedures	11
Test Connectors	13

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

LIST OF ILLUSTRATIONS

Figure 1. DID0-32A ABUS 2nd LEVEL	2
Figure 2. Second Level Logics of 32 Bit DID0 Macrofunction Card	4
Figure 3. Digital Input Circuit	5
Figure 4. Digital Output Circuit	6
Figure 5. Register Read Timing Diagram	6
Figure 6. Register Write Timing Diagram with Times	7
Figure 7. User Pin Definitions and Assignments	8
Figure 8. Digital Input Hardware Programming	9
Figure 9. Digital Input Isolation	11
Figure 10. Output Isolation	12

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

32 BIT DI/DO CARD

GENERAL DESCRIPTION

Abstract

The 32 bit DI/DO card is implemented as a macro-function card in the RDIS Ariel (RDIS-A). This card will allow a maximum of sixteen digital inputs (DI's) along with sixteen digital outputs (DO's) to the user. The design utilizes low power VTL along with the reliability of IBM released GOLF and COMPASS technology modules. All digital inputs are hardware programmable as EIA or simple Digital Inputs. The digital outputs are hardware programmable as active collector or open collector with 64 ma maximum drive/sink capability. Access to critical points for debug purposes has been added increasing servicability.

Features

- Sixteen Digital Inputs
 - EIA standard 75154 receivers, or Vtl receivers
 - GND is provided for isolation purposes
- Sixteen Digital Outputs
 - Hardware programmable open/active collector drivers
 - 64 ma sink capability of at most 8 DO at any one time
 - 25 ma sink capability of all DOs at any one time
 - GND is provided for isolation purposes
 - Readback of output registers for increased reliability
- Miscellaneous
 - The Macro-Function ID is hex '30'
 - +5 volts at 1 amp/card is supplied from the power supply for isolation purposes
 - Simple to use hardware programming pins at both the output and input
 - LED to signal card select
 - Parity is checked on all DIS write commands and generated on all DIS read commands
 - The input register clocks are held off during read operations to insure data integrity
 - Direct interrupt line from the user to the controller

IBM Internal Use Only

Part Numbers and Related Documents

Part numbers referring to hardware are listed below:

1. Assembly - 6912757
2. Raw Card - 6912758
3. Schematic - 6912759
4. E.C. - A37943F

Related Documents

1. Distributed Sensor Subsystem IV Functional Spec E45-1289-0 PN 783492
2. Distributed Sensor Subsystem User Guide (Volume I), Z45-1123, PN 6856716
3. Distributed Sensor Subsystem User Guide (Volume II), Z45-1124, PN 4745524
4. Remote DIS Maintenance Manual
5. Remote DIS Functional Spec E45-1353, PN 6015823

A-BUS INTERFACE DEFINITION

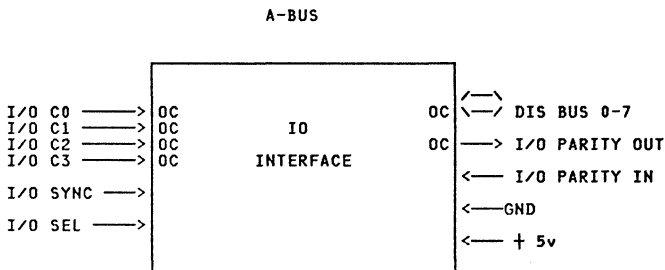


Figure 1. DIDO-32A ABUS 2nd LEVEL: The DIS channel is an open-collector bus. The terminators for the OC signals will reside on the board.

A-BUS Inputs and Outputs

The A-Bus will provide the path for data transfer to/from the DIDO-32A card. All control, Data Busses, Address, and Power will be provided from the A-Bus.

The following is a list of all the A-Bus I/O on DIDO-32A and their definitions as a function of the Hazel-A card. These I/O are shown in Figure 1.

- IO SYNC I DIS Sync ; When active, signals data and CTC's are valid on the DIS Bus to the DIDO-32A card provided IO Select is active.

IBM Internal Use Only

- **IO RETURN IN** **O** DIS Return ; When reading DIDO-32A, this output signals data is valid on the bus. When writing DIDO-32A, this output acknowledges the data has been received and is valid, (ie. good parity). It is enabled when Sync, and IO Sel are active during a DIS Read, and Sync, IO Sel, and parity is valid during a DIS Write provided a valid ctc has been sent.

- **IO SELECT** **I** DIS LSA Select ; This input Enables DIDO-32A. When active, it will allow the DIS Bus, C0-C3, Sync, and Return through it's buffers.

- **C0,C1,C2,C3** **I** Command Tag Combination ; These inputs are used as control signals to DIDO-32A. They are valid when IO Sel, and Sync are valid.

- IO D0-D7** **I/O** DIS Bus Bits 0-7 ; These bidirectional open-collector signals are the data bus to/from the DIDO-32A card. When reading DIDO-32A, data is valid before Return becomes active. When writing DIDO-32A, data is valid before Sync becomes active.

- IO DP IN** **O** DIS Data Parity IN ; This output is generated to guarantee Negative Active ODD Parity across the DIS Bus (ie. an odd number of gnnds on the DIS Bus and IO DP IN)

- IO DP OUT** **I** DIS Data Parity Out ; This input and the DIS Bus are checked to guarantee Negative Active ODD Parity during a DIS Write. If Parity is bad (ie. not an odd number of gnnds across the 9 bits), then return will not be generated.

Voltages **I** There are 9 different voltage levels available on the Abus connector. They are;

Voltages		Pins	
1.	gnd	116, 118, 120	
2.	+ 5v	115, 117, 119	1.5 A max
3.			
4.	- 5v	71	not used
5.	+ 8.5v	33	not used
6.	+ 12v	48	not used
7.	- 12v	94	not used
8.	+ 15v	1	not used
9.	- 15v	89	not used
10.	Agnd	2	not used

Note: Agnd is Analog ground and should be kept separate from gnd (digital ground).

SECOND LEVEL LOGICS

The block diagram illustrates the internal architecture of the GOLF module. At the top, control signals C0, C1, C2, C3, LSA SEL, and SYNC are connected to an RCVR (Receiver), which then feeds into a DECODER and a NAND gate. The NAND gate's output goes to an AND gate, which produces the RTN (Return) signal. The RCVR also provides a signal 'A' to the CONTROL LOGIC. The CONTROL LOGIC is a central hub that manages data flow between several components: four EIA/VTL REC invert blocks (receiving DI 0-3, 4-7, 8-11, and 12-15), two 8-bit REG (Register) blocks, a third REG block, a PAR CHKR (Parity Checker), and a FUN (Function) block. The EIA/VTL REC invert blocks output 4-bit data to the first two REG blocks. The first REG block outputs 8-bit data to the second REG block. The second REG block outputs 8-bit data to the third REG block. The third REG block outputs 8-bit data to the FUN block. The PAR CHKR receives 8-bit data from the first REG block and outputs a signal to the RCVR. The RCVR also receives data from the DIS BUS (9). The GOLF MODULE is connected to the external world via a 330 ohm resistor, which provides contact and cleaning current. This resistor is connected to a 6.8 uF capacitor and a 3.6k ohm resistor, which in turn connects to the INTERRUPT (proc) signal. The INTERRUPT (user) signal is also connected to the 3.6k ohm resistor and the 6.8 uF capacitor. The 3.6k ohm resistor is labeled as a debounce circuit.

32 Bit DI/DO Card

IBM Internal Use Only

Digital input signals are received through EIA receivers and latched all the time, except when the card is being read. In this case the data input register's clocks are turned off. The Digital outputs are always enabled and are read through the DIS Bus along with the Digital inputs.

Parity is checked during DIS writes and generated for DIS reads.

HARDWARE

The digital input circuit provides capability to receive standard RS232C EIA inputs or vtl inputs. The inputs are gated through 4 EIA receivers each with their own program jumper. If the jumper is on, the receiver can accept only EIA signals. If the jumper is off, the receiver can accept only VTL inputs. See Figure 3.

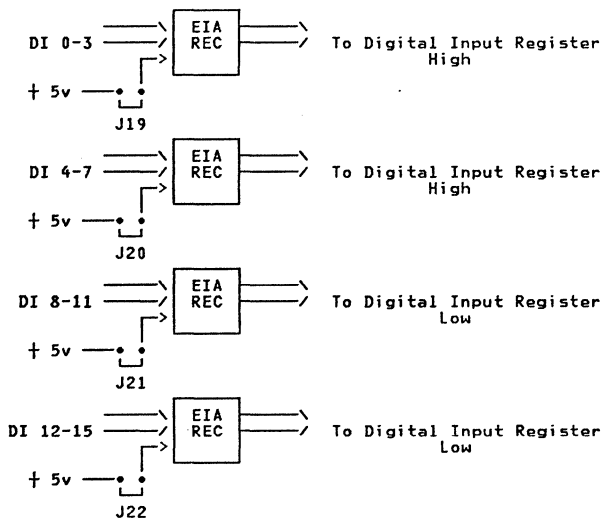


Figure 3. Digital Input Circuit

The digital outputs are driven through 64 ma drivers with jumpers for each. If the jumper is off, the load resistor (pull-up) on the digital output can be selected to sink 64 ma. If the jumper is on, the max load resistance cannot allow more than 29 ma ($(64\text{ma} - (5/200)\text{ma}) - 10\text{ma}$). The 10 ma is a safety margin for the 32 DI/DO32-A card. Not more than EIGHT digital outputs can sink 64 ma at any one time. But all 16 DOs can sink 30 ma at any given time. See Figure 4 on page 6

IBM Internal Use Only

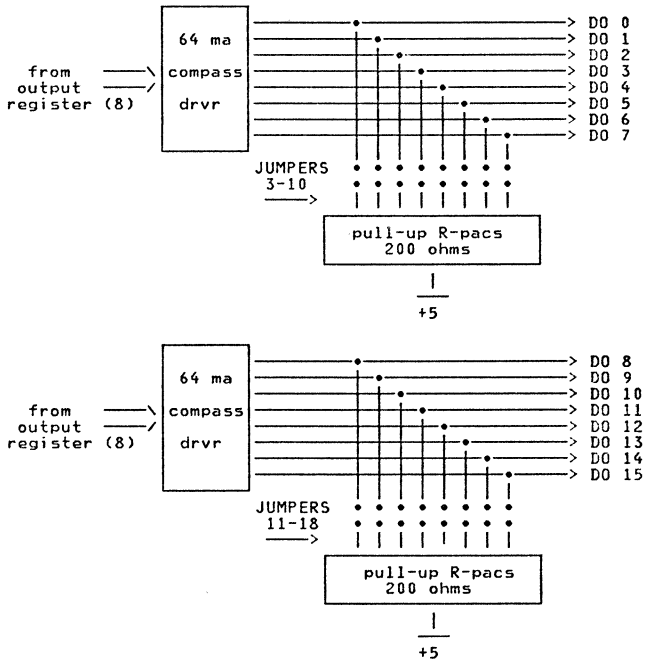


Figure 4. Digital Output Circuit

TIMING

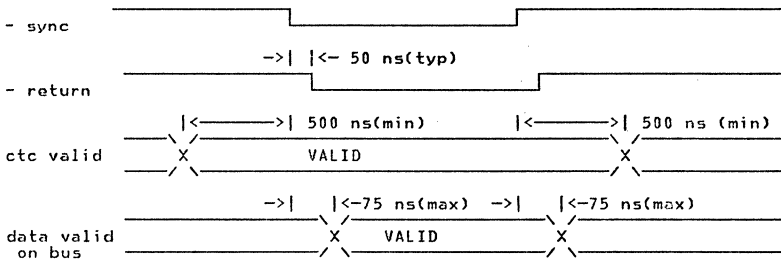


Figure 5. Register Read Timing Diagram

IBM Internal Use Only

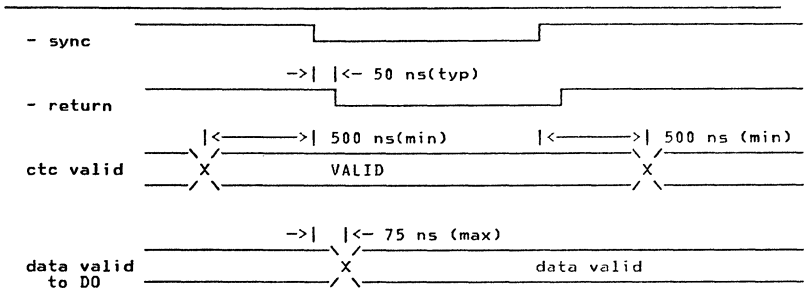


Figure 6. Register Write Timing Diagram with Times: Parity is checked on a DIS write. If parity is bad then return is blocked to the processor.

COMMAND TAG DEFINITIONS

CTC 0	LSA Select (I/O Sel)
CTC 4	Read DI Reg High (0-7)
CTC 5	Read DI Reg Low (8-15)
CTC 6	Read D0 Reg High (0-7)
CTC 7	Read D0 Reg Low (8-15)
CTC A	Write D0 Reg High (0-7)
CTC B	Write D0 Reg Low (8-15)
CTC F	Read Macro-Function ID
CTC 1	not used
CTC 2	not used
CTC 3	not used
CTC 8	not used
CTC 9	not used
CTC C	not used
CTC D	not used
CTC E	not used

IBM Internal Use Only

I/O

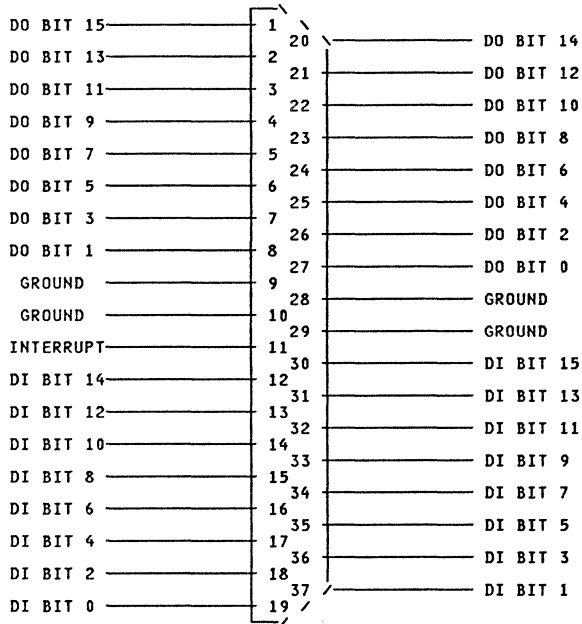


Figure 7. User Pin Definitions and Assignments: This is a 37 pin d-shell IBM P/N 6149157, AMP P/N 745256-1

IBM Internal Use Only

HARDWARE PROGRAM JUMPERS

Jumper	Jumper Position	Function	DI Bits
19	on	EIA receiver	0-3
20	on	EIA receiver	4-7
21	on	EIA receiver	8-11
22	on	EIA receiver	12-15
19	off	VTL receiver	0-3
20	off	VTL receiver	4-7
21	off	VTL receiver	8-11
22	off	VTL receiver	12-15

Figure 8. Digital Input Hardware Programming: If the jumper is off the receiver can accept both EIA, and VTL. But if in VTL mode and input is EIA, the data being read may be invalid.

Note: See 'The Line Driver and Line Receiver Data Book' from Texas Instruments (75154 quadruple line receiver) for details.

VTL Mode Threshold Voltages

low -> + 1.4 volts
high -> + 2.2 volts

EIA Mode Threshold Voltages

low -> - 1.2 volts
high -> + 2.2 volts

IBM Internal Use Only

Programming the Digital Outputs

Jumper	Jumper Position	Function	DO Bit
3	on	active collector	0
4	on	active collector	1
5	on	active collector	2
6	on	active collector	3
7	on	active collector	4
8	on	active collector	5
9	on	active collector	6
10	on	active collector	7
11	on	active collector	8
12	on	active collector	9
13	on	active collector	10
14	on	active collector	11
15	on	active collector	12
16	on	active collector	13
17	on	active collector	14
18	on	active collector	15
3	off	open collector	0
4	off	open collector	1
5	off	open collector	2
6	off	open collector	3
7	off	open collector	4
8	off	open collector	5
9	off	open collector	6
10	off	open collector	7
11	off	open collector	8
12	off	open collector	9
13	off	open collector	10
14	off	open collector	11
15	off	open collector	12
16	off	open collector	13
17	off	open collector	14
18	off	open collector	15

IBM Internal Use Only

MISCELLANEOUS HARDWARE ISOLATION PROCEDURES

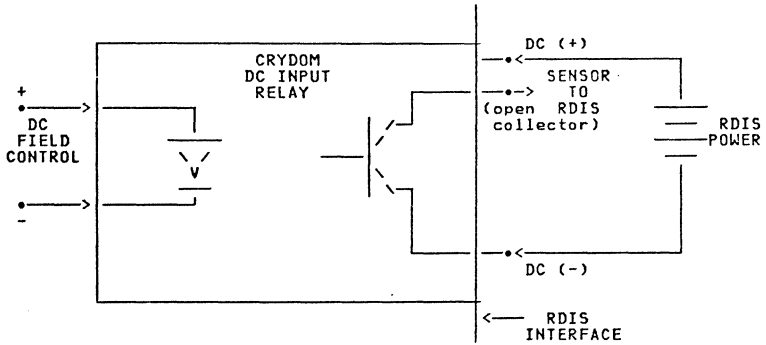
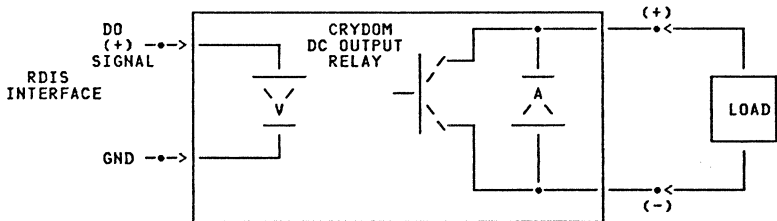


Figure 9. Digital Input Isolation: This figure is only an example of how the digital inputs could be isolated

IBM Internal Use Only

+ Active Collector Digital Output

Output Isolation for a plus Active Digital Output



Output Isolation for a minus Active Digital Output

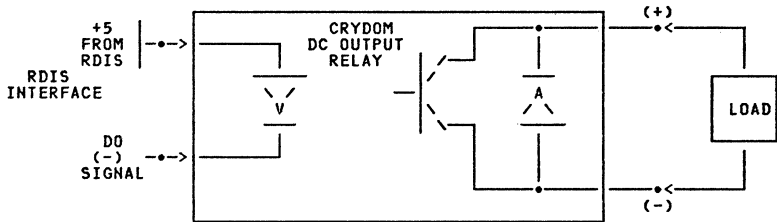


Figure 10. Output Isolation: Power & Gnd is taken from a user output from the power supply Gnd is also available on the 37 pin D-shell.

IBM Internal Use Only

TEST CONNECTORS

Test Point	Description	Location	Active State
0	- CTC 4	Z8 pin 13	low/gnd
1	- CTC 5	Z8 pin 11	low/gnd
2	- CTC 6	Z8 pin 10	low/gnd
3	- CTC 7	Z8 pin 9	low/gnd
4	- CTC A	Z8 pin 6	low/gnd
5	- CTC B	Z8 pin 5	low/gnd
6	- CTC F	Z8 pin 1	low/gnd
7	- MACRO ENABLE	Z14 pin 3	low/gnd
8	+ VALID CTC	Z9 pin 8	high/+5v
9	+ GATE SYNC	Z7 pin 6	high/+5v
10	- WR TO XREG	Z7 pin 12	low/gnd
11	- MACRO ID	Z7 pin 8	low/gnd
12	GND		low/gnd
13	+ 5 VOLTS		high/+5v
14	BUF DIS BUS BP	Z16 pin C01	low/gnd
15	BUF DIS BUS B7	Z16 pin C03	low/gnd
16	BUF DIS BUS B6	Z16 pin A01	low/gnd
17	BUF DIS BUS B5	Z16 pin C05	low/gnd
18	BUF DIS BUS B4	Z16 pin A09	low/gnd
19	BUF DIS BUS B3	Z16 pin E09	low/gnd
20	BUF DIS BUS B2	Z16 pin J07	low/gnd
21	BUF DIS BUS B1	Z16 pin J03	low/gnd
22	BUF DIS BUS B0	Z16 pin J01	low/gnd
23	+ PAR ERROR	Z13 pin 6	high/+5v

Heading ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Heading References</u>
abusio	DI032A	2	A-BUS Inputs and Outputs
tp	DI032A	13	Test Connectors

Figure ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Figure References</u>
habuss	DI032A	2	1: 2
second	DI032A	4	2:
input	DI032A	5	3: 5
output	DI032A	6	4: 5
regrd	DI032A	6	5:
regwr	DI032A	7	6:
io	DI032A	8	7:
iniso	DI032A	11	9:
plus	DI032A	12	10:

Imbed Trace

**Ariel System
Analog to Digital Converter Card**

Control Systems Engineering

**IBM Corporation
General Technology Division
D-035 B-300-47A
East Fishkill
tel 533-7654**

IBM Internal Use Only

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

TABLE OF CONTENTS

General Description	1
Abstract	1
Features	2
Power Requirements	2
Logic Conventions	2
Part Number Information	2
Second Level Diagram	3
Hardware Description	5
Module Specifications	5
Analog to Digital Converter	5
Instrumentation Amplifier	5
Current Input Bridge	5
A/D DIS Interface Bus	6
Pin Definitions (120 pin)	7
Timing Diagram	8
CTC Assignment	9
CTC Descriptions	9
Truth Table for Analog Channel Select	10
Truth Table for Gain Select	10
Additional Information for Operation	10
User Pin Assignments	11
Customer Interface Connector	11
Current loop resistor bridge	12
Test Points	12
Program Points	12
A/D Converter Mode Of Operation/Selection	12
Binary Conversion Guides to Selected Mode	14
Bipolar Mode, 20 volt Span selected:	14
Bipolar Mode, 10 volt Span selected:	14
Unipolar Mode, 10 volt Span selected:	15
Gain Selection (Jumper)	15
Calibration Procedure	17

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

LIST OF ILLUSTRATIONS

Figure 1. RDIS ADC Second Level Diagram	3
Figure 2. A/D DIS Bus	6
Figure 3. Customer Interface Connector	11
Figure 4. Trim pots for A/D calibration	17

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

GENERAL DESCRIPTION

ABSTRACT

The Ariel Remote Distributed Interface System (RDIS) Analog to Digital Converter (ADC) card is a sixteen channel, differential input, programmable gain, data acquisition sub-system designed to interface to the East Fishkill Distributed Interface System (DIS). The card can accept voltage or 0-20 ma current input (for current input the user must install a resistor per channel to an on-board bridge). Vital modules are socketed and test points are brought to the card edge for maintenance. The design utilizes analog and digital VTL modules as well as IBM Golf Technology modules.

IBM Internal Use Only

FEATURES

- 16 analog input channels (37 pin D shell)
- 12 bit resolution
- Programmable gain and channel select using ctc
- Jumper selected gain and input ranges
- LED to signal card selection
- Resistor bridge to accept 4-20ma current input
- Test pins easily accessible at card edge
- +10V precision reference to user interface
- External start capability
- Interrupt on end-of-conversion
- Parity check and generation across DIS data channel
- Compatible with standard DIS interface

POWER REQUIREMENTS

A regulated power supply should be used to provide the following voltages:

- ± 15 volts(± 5 %)at 400 mA each
- +5 volts @ 1 A

LOGIC CONVENTIONS

Logic busses to and from the BIC logic are negative logic convention ($-=1,+=0$). Internal logic on the card is positive logic ($-=0,+=1$).

PART NUMBER INFORMATION

- Raw Card - Part Number 6912766
- Schematic - Part Number 6912767
- Assembly - Part Number 6912768

IBM Internal Use Only

SECOND LEVEL DIAGRAM

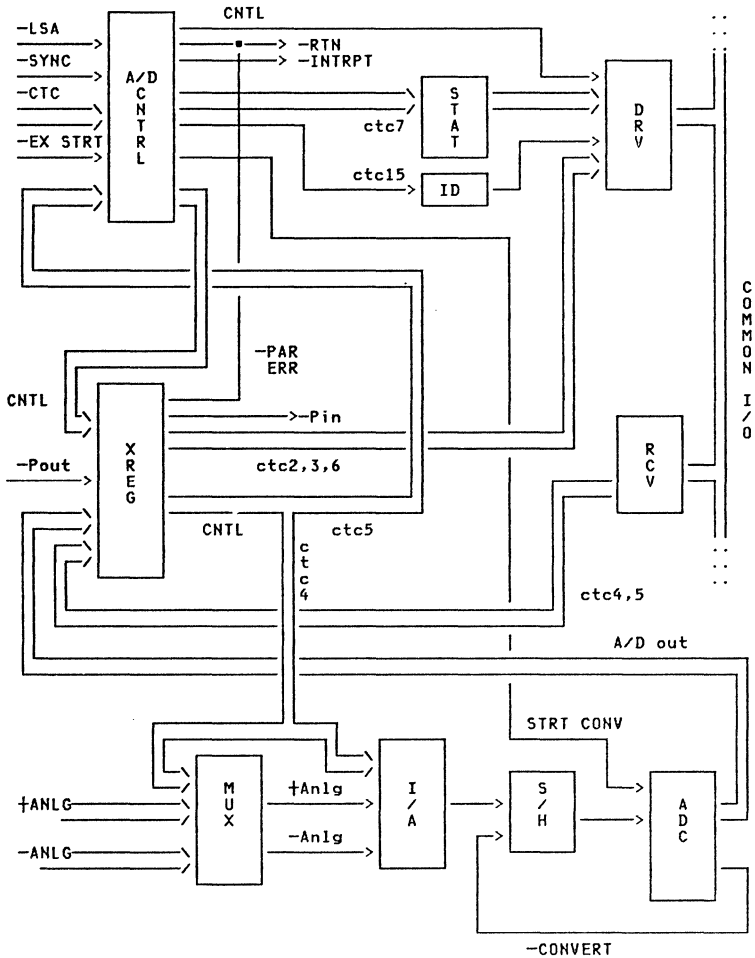


Figure 1. RDIS ADC Second Level Diagram

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

HARDWARE DESCRIPTION

MODULE SPECIFICATIONS

Analog to Digital Converter

- Maximum input voltage of ± 10 volts
- 12 bit output
- Conversion time of 25 microseconds
- Low power (900mw)
- Input ranges jumper selected ($\pm 5, \pm 10, 0-10$)

Instrumentation Amplifier

- Input stage programmable to gains of 1, 2, 5, or 10
- Output stage jumper selected gains of 1, 4, or 10
- Input voltage x gain not to exceed 10 volts
- Discrete zeners on board to protect amplifier input

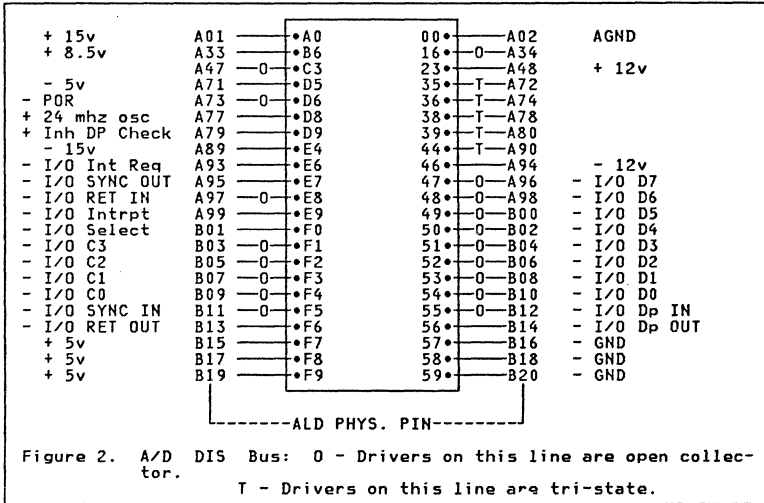
CURRENT INPUT BRIDGE

A resistor bridge is located on the right section of the board. To allow for a 0-20ma channel simply solder a resistor across the appropriate slots for that channel. A 250 ohm resistor will convert 0-20ma to 0-5V. A precision resistor is being released for access to anyone who requires it :

.01% precision
250 ohms
1/8 watt
50 ppm/C
part number: 55X0215

IBM Internal Use Only

A/D DIS INTERFACE BUS



IBM Internal Use Only

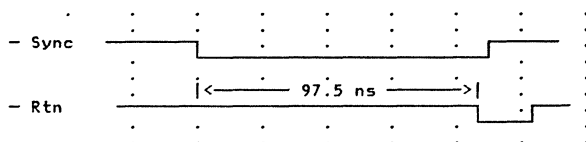
Pin Definitions (120 pin)

-POR	Power On Reset. unused by A/D.
-I/O D0-D7	Bidirectional I/O Data Bus. 8-bit bidirectional data bus to/from AP/SCA or DCA cards.
-I/O Dp OUT	I/O Data Parity Out. Odd parity bit for I/O D0-D7 from BIC logic to this card, valid during an I/O write operation.
-I/O Dp IN	I/O Data Parity In. Odd parity bit for I/O D0-D7 from this card to BIC logic, valid during an I/O read operation.
-I/O C0-C3	These are the 4 command tag lines representing the command to this card.
-I/O SYNC OUT	I/O SYNC Out. Control line generated by BIC logic to initiate a transaction on the I/O bus. When low, data and command tag information is valid for processing.
-I/O RET IN	I/O RETURN In. Control line generated by this card to signal the successful completion of a transaction on the I/O bus. When low, return data is valid for processing by the BIC logic.
-I/O Select	I/O LSA Select. This line is used by this macrofunction card to mean it has been selected and that -sync is valid.
-I/O Intrpt	I/O LSA Interrupt. This line is an input to one of the bits in the BIC Interrupt Register and is used by this card to indicate that the A/D converter has successfully made a conversion. This line is active when it is at a minus level.

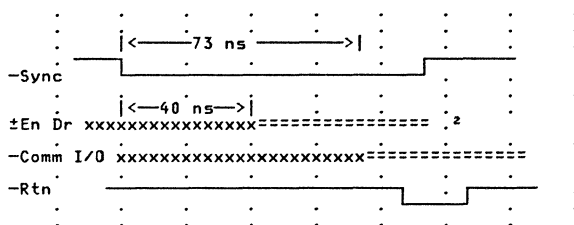
IBM Internal Use Only

TIMING DIAGRAM

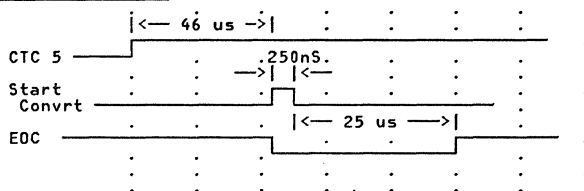
DIS Write¹



DIS Read¹



Conversion Cycle³



¹ Time scale for DIS read and write timing diagrams is 20 ns per interval.

² xx means data is in don't care condition

³ Time scale for the conversion cycle is 2⁰ us per interval.

IBM Internal Use Only

CTC ASSIGNMENT

CTC DESCRIPTIONS

CTC	Function	Data Transferred
0	LSA Select	No Data Transferred
1	Rst Intr Req, Intr Pend, and Diag Latch	No Data Transferred
2	Read Data Reg High	Bit 0 EOC 1-3 Don't Care 4 Bit 1(MSB, Sign bit) 5 Bit 2 6 Bit 3 7 Bit 4
3	Read Data Reg Low	8 Bit 5 : 15 Bit 12(LSB)
4	Write Control Registers High	Bit 0-3 Select Analog Channel 4-5 Gain Select
5	Write Control Register Low	Bit 8 Start Convert 9 Enable Interrupt 10 Allow Ext Start
6	Read Status Registers Hi	Bit 0-3 Sel Analog Chnl Bit 4-5 Sel Inst Amp Gain
7	Diag Off: Read Status Register Low Diag On:	Bit 8 Intr Pending 9 E-O-C 10 Intr Enabled 11 Ext Start Allowed Data from last ctc 5
9	Set Diagnostic Latch	No Data Transferred
D	Reset Data Control	No Data Transferred
F	Read Card ID	X'2C'

IBM Internal Use Only

TRUTH TABLE FOR ANALOG CHANNEL SELECT

ctc 4 bits 0-3	chan sel
1111	0
1110	1
1101	2
1100	3
1011	4
1010	5
1001	6
1000	7
0111	8
0110	9
0101	10
0100	11
0011	12
0010	13
0001	14
0000	15

TRUTH TABLE FOR GAIN SELECT

Programmable Gain Selection		
D0 (Bit 4)	D1 (Bit 5)	GAIN
1	1	1
1	0	2
0	1	5
0	0	10

ADDITIONAL INFORMATION FOR OPERATION

Upon power on, the initializing sequence for the card should be a ctc D followed by a ctc 1.

Parity is checked and generated across the 8 DIS data bits. When bad parity is detected, the Return signal is inhibited. However, the Start Convert signal is not inhibited. Thus an interrupt (conversion complete) should be ignored when a ctc error (no return from card) is detected.

When in the external mode, the external start signal must be -active with a minimum pulse of 1ns.

When changing the level of the control bits in the low register, do a ctc 5 with all zeroes followed by ctc 1 before writing the new data to the register (ctc 5). This will be required, for example, when going from 'interrupt enabled' to 'interrupt not enabled'.

IBM Internal Use Only

USER PIN ASSIGNMENTS

CUSTOMER INTERFACE CONNECTOR

The customer interface connector is a 37 pin 'D' shell connector mounted on the edge of the card. This connector receives sixteen analog signals and the external start. Also supplied to the user are the 10V reference and analog ground.

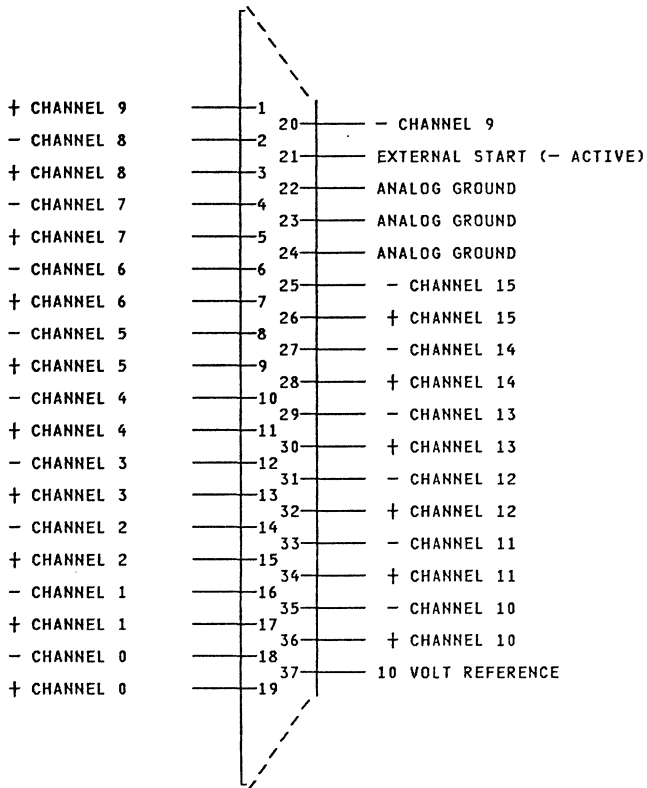


Figure 3. Customer Interface Connector

It is recommended to use shielded twisted pair for analog input lines over 10 feet. The shield should be grounded at the source end. In noisy environments, the maximum gain used should not exceed 5.

IBM Internal Use Only

CURRENT LOOP RESISTOR BRIDGE

Located near the 'd' shell connector is a 36 pin DIP solder bridge. the bridge comes mounted into a 36 pin socket so it is removable. See assembly drawing 6912766 for more details.

The bridge is used if it is desired to change the input signal from voltage mode to current loop mode. All the input signals from the 'd' shell are also wired to the bridge to pairs of contacts. Placing a resistor across these contacts is all that is required to sense loop current. Resistor values are calculated by knowing the loop current upper limit at 5 volts, e.g. the calculated value for a 4-20 ma current loop is 250 ohms. It is recommended that all resistors should be precision .01%.

The bridge socket labels the contact pairs as channel 0-15 for ease of resistor installation. The fact that the resistor bridge is removable makes it possible for bridges to be pre-made and stocked or removed and used on a replacement of a defective card.

TEST POINTS

The following is a list of the test points accessible along the upper edge of the RDIS ADC card:

TEST POINT	DESCRIPTION
1	-Return
2	-Enable Interrupt
3	-Start Convert
4	-Enable Drivers
5	+Enable Drivers
6	+Analog Signal In
7	-Analog Signal In
8	Analog Out (of Instr. Amp.)
9	Analog Out (of Sample/Hold)
10	Analog Ground
11	-Converting
12	+Converting
13	+10 V Precision Reference
14	4 MHz Clock

PROGRAM POINTS

A/D Converter Mode Of Operation/Selection

Mode of Operation		
	Bipolar	Unipolar
10 Volt Span	± 5 Volts	0 - 10 Volts
20 Volt Span	± 10 Volts	Not Allowed

After selecting one of the above modes, use the following table to place the appropriate jumpers on the card.

IBM Internal Use Only

Mode Selection								
	Bipolar	Unipolar						
10 Volts	<table border="1"><tr><td>J</td><td>J</td><td></td></tr></table>	J	J		<table border="1"><tr><td></td><td>J</td><td></td></tr></table>		J	
J	J							
	J							
20 Volts	<table border="1"><tr><td>J</td><td></td><td>J</td></tr></table>	J		J	Not Allowed			
J		J						

The jumper bank shown in the above chart is indicated on the card as follows:

A

--	--	--

(Lower left portion of card)

BIP 10 20
SPAN

IBM Internal Use Only

Binary Conversion Guides to Selected Mode

The following charts show ADC outputs for the three input ranges at gain equal to 1. The analog signal must not exceed the maximum range divided by the gain. The following formula may be used to convert the output of the ADC to the input voltage.

$$\text{(LSB/Gain)} \times \text{Output of A/D}$$

(where LSB is at gain=1)

Bipolar Mode, 20 volt Span selected:

LSB = 0.0049 volts
Gain = 1

FORM= 2's comp

Voltage	Digital Output
+9.9951 v	0 1 1 1 1 1 1 1 1 1
+9.9902 v	0 1 1 1 1 1 1 1 1 0
.	.
+0.0049 v	0 0 0 0 0 0 0 0 0 1
+0.0000 v	0 0 0 0 0 0 0 0 0 0
-0.0049 v	1 1 1 1 1 1 1 1 1 1
-9.9951 v	1 0 0 0 0 0 0 0 0 1
-10.0000 v	1 0 0 0 0 0 0 0 0 0

Bipolar Mode, 10 volt Span selected:

LSB = 0.0024 volts
Gain = 1

FORM= 2's comp

Voltage	Digital Output
+4.9976 v	0 1 1 1 1 1 1 1 1 1
+4.9952 v	0 1 1 1 1 1 1 1 1 0
.	.
+0.0024 v	0 0 0 0 0 0 0 0 0 1
+0.0000 v	0 0 0 0 0 0 0 0 0 0
-0.0024 v	1 1 1 1 1 1 1 1 1 1
-4.9976 v	1 0 0 0 0 0 0 0 0 1
-5.0000 v	1 0 0 0 0 0 0 0 0 0

IBM Internal Use Only

Unipolar Mode, 10 volt Span selected:

LSB = 0.0024 volts

Gain = 1

FORM= binary weight with MSB inverted

Voltage	Digital Output
+9.9976 v	0 1 1 1 1 1 1 1 1 1
+9.9952 v	0 1 1 1 1 1 1 1 1 0
.	.
+5.0024 v	0 0 0 0 0 0 0 0 0 1
+5.0000 v	0 0 0 0 0 0 0 0 0 0
+4.9976 v	1 1 1 1 1 1 1 1 1 1
.	.
+0.0024 v	1 0 0 0 0 0 0 0 0 1
+0.0000 v	1 0 0 0 0 0 0 0 0 0

Gain Selection (Jumper)

The hardware gain is selected by placing jumpers on the appropriate jumper banks as follows:

B	X10	X4	X1
---	-----	----	----

**Both rows must have
one jumper only.
Banks located on lower
left portion of card

C	X10	X4	X1
---	-----	----	----

**Note: This is a multiplicative gain with the
programmed gain.

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

CALIBRATION PROCEDURE

The sequence in which the analog modules are to be trimmed for offset and gain is as follows: the instrumentation amplifier, the sample and hold amplifier, and last, the A/D converter. The Instrumentation Amplifier and the Sample/Hold Amplifier need to be adjusted so the offset is zero, and the Analog to Digital Converter has to be adjusted for zero and gain. Zero is always adjusted first and then the gain. Zero is adjusted when the analog input is near the most negative end of the analog range, and gain is adjusted with the analog input near the most positive end of the analog range.

First-stage gain is selected by issuing a CTC 4 with proper gain select settings, and second-stage gain is selected via jumpers. The instrumentation amp should be re-zeroed each time the second-stage gain is adjusted, for maximum accuracy.

1. Select gain jumpers
2. Select a power source that is capable of producing input from -10 volts to +10 volts.
3. Use ctc 4 to select the channel where the signal is connected to and then followed by a ctc 5 with start convert on to enable the channel selection.
4. Set the input so that the differential input is zero volt.
5. Set the 10 k-ohm pot to center.
6. Set the first-stage gain to 1. Measure output of instrumentation amplifier, Vout1.
7. Set the first-stage gain to 10. Measure output voltage, Vout2.
8. Calculate portion of Vout2 contributed by the output stage offset by

$$Voos=1/9(10 \times Vout1 - Vout2)$$

9. Adjust the 10 k-ohm pot until the output voltage is equal to the calculated Voos.
10. Adjust the 1 k-ohm pot(see Figure 4) so the output reads zero.
11. Set the reference input to -9.9952 volts (taking gain settings into consideration) and issue a ctc 5 with start convert on, read the output using ctc 2 and 3.
12. Repeat the above step and adjust the 20 k-ohm pot(see Figure 4) until the digital output is 1000 0000 0001.
13. Now, set the input to +9.9902 volts and start a conversion with ctc 5.
14. Repeat the above step and adjust the 100 ohm pot until the output of the A/D is 0111 1111 1110.

Note: Steps 11-14 are for $\pm 10V$ range. Adjust the input accordingly if using another range.(See Binary Conversion Guide in this document).

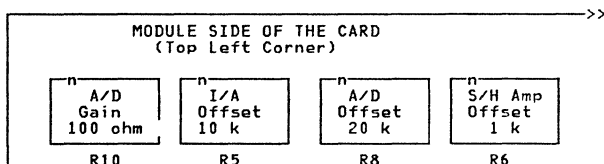


Figure 4. Trim pots for A/D calibration

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

Figure ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Figure References</u>
slref	ALSADC	3	1:
abus	ALSADC	6	2:
dshell	ALSADC	11	3:
xpotref	ALSADC	17	4: 17, 17

Footnote ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Footnote References</u>
disref	ALSADC	8	1: 8, 8
xxxref	ALSADC	8	2: 8
ccref	ALSADC	8	3: 8

**Ariel System
Digital to Analog Converter Card**

Control Systems Engineering

**IBM Corporation
General Technology Division
D-035 B-300-47A
East Fishkill
tel 533-7654**

IBM Internal Use Only

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

TABLE OF CONTENTS

General Description	1
Abstract	1
Features	2
Power Requirements	2
Part Number Information	2
Logic Conventions	2
DAC Card Second Level Diagrams	3
Data Flow	3
Analog Output Circuit	3
DAC DIS Bus	5
Pin Definitions (120 pin)	6
Hardware Description	7
Theory Of Operation	7
DAC Module	7
DAC Data Format	7
Output Range Selection	8
CTC Assignment	9
Command Descriptions	9
Command Tag Summary	9
Command Sequence Examples	10
Normal Operation	10
Diagnostic Operation	10
User Pin Assignments	11
Customer Interface Connector	11
Test Points	11
Calibration Procedure	13
Procedure	13

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

LIST OF ILLUSTRATIONS

Figure 1.	Data Flow	3
Figure 2.	Analog Output Circuit	4
Figure 3.	ARIEL DIS Bus	5
Figure 4.	DAC Card Analog Output vs Digital Input	8
Figure 5.	DAC Data Conversion Equation and Constants	8
Figure 6.	Jumper Positions and Output Ranges	8
Figure 7.	Customer Interface Connector	11
Figure 8.	Digital Test Points	12
Figure 9.	Analog Test Points	12

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

GENERAL DESCRIPTION

ABSTRACT

The Ariel Remote Distributed Interface System (RDIS) Digital to Analog Converter (DAC) card, is an eight channel analog output card designed to interface to the Distributed Interface System (DIS). The Command Tag Combination (CTC) assignment is identical to the standard (DIS) Multi-Level DAC Card during normal operation. This card has eight voltage outputs with ranges independently selected via board jumpers. The card is implemented using TTL-LS logic and a compact, quad, twelve bit, internally latched DAC module. Test pins are provided on the card to enhance servicability.

IBM Internal Use Only

FEATURES

- 8 Digital to Analog Channels
- 12 Bit Resolution
- 5 Jumper Selectable Voltage Ranges per Channel
- Gain and Offset Adjustments on Each Channel
- Quad DAC's and Op Amps Save Space
- TTL-LS Modules Save Power
- Test Pins for Diagnostics
- Socketed DAC's for Maintainability
- Software Compatible With the Existing DIS Multi Level DAC

POWER REQUIREMENTS

VOLTAGE	TYP POWER (ma)	MAX POWER (ma)
+5	280	485
+15	45	75
-15	175	220

PART NUMBER INFORMATION

- Assembly 6912769
- Raw Card 6912770
- Schematic 6912771

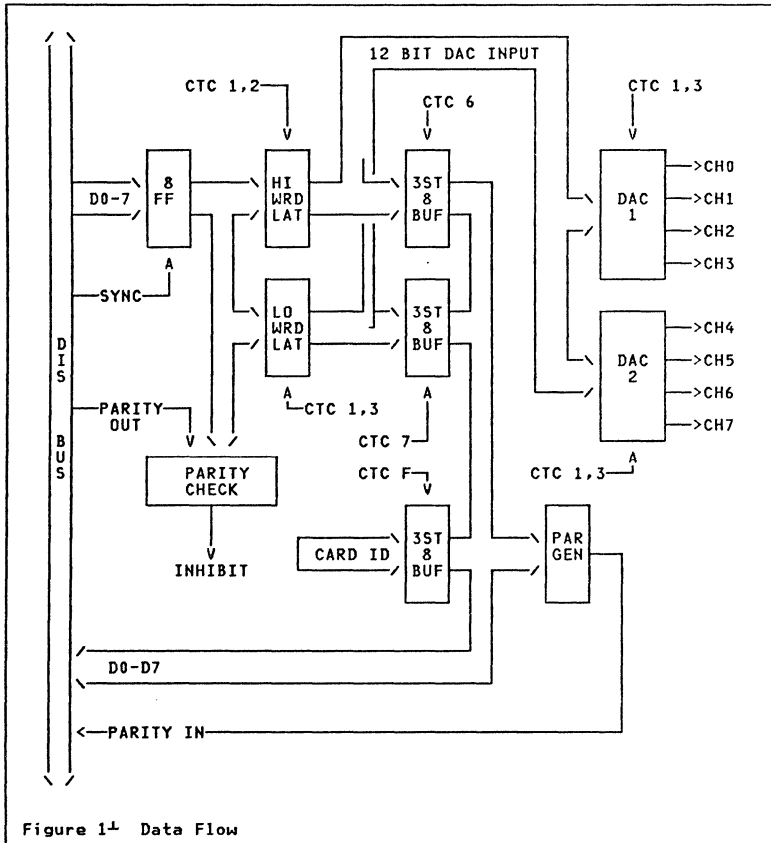
LOGIC CONVENTIONS

Logic busses to and from the Block Interface (BIC) logic are negative logic convention, (-=1, +=0). Internal logic on the card is positive logic (-=0, +=1).

IBM Internal Use Only

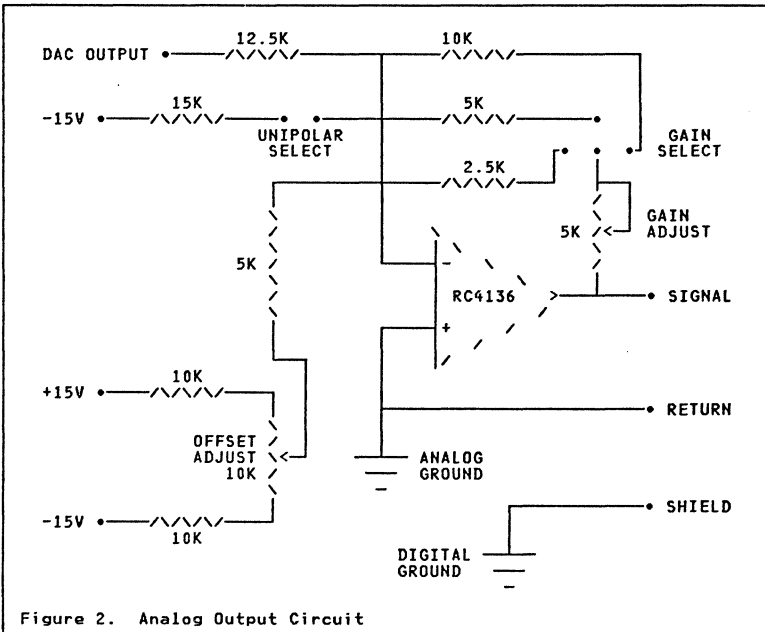
DAC CARD SECOND LEVEL DIAGRAMS

DATA FLOW



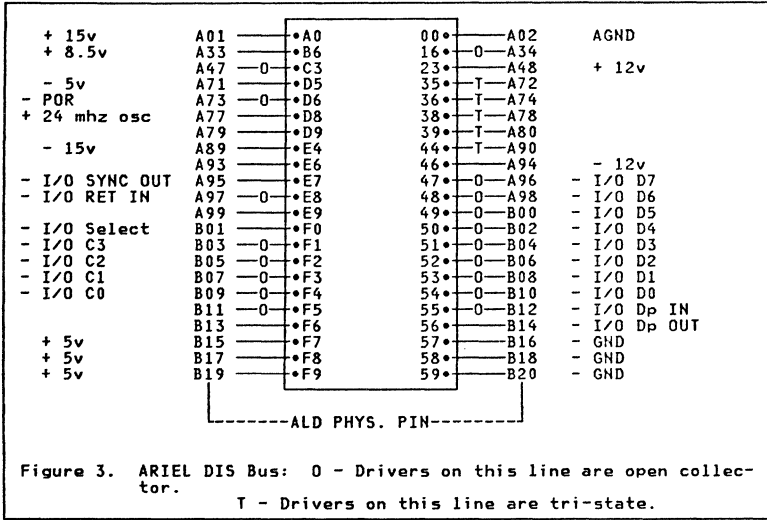
ANALOG OUTPUT CIRCUIT

IBM Internal Use Only



IBM Internal Use Only

DAC DIS BUS



IBM Internal Use Only

Pin Definitions (120 pin)

-I/O D0-D7	Bidirectional I/O Data Bus. 8-bit bidirectional data bus to/from macrofunction cards.
-I/O Dp OUT	I/O Data Parity Out. Odd parity bit for I/O D0-D7 from BIC logic to this card, valid during an I/O write operation.
-I/O Dp IN	I/O Data Parity In. Odd parity bit for I/O D0-D7 from this card to BIC logic, valid during an I/O read operation.
-I/O C0-C3	Bidirectional I/O Command Tag Bus. 4-bit bidirectional command tag bus to/from macrofunction cards.
-I/O SYNC OUT	I/O SYNC Out. Control line generated by BIC logic to initiate a transaction on the I/O bus. When low, data and command tag information is valid for processing if this card is currently selected.
-I/O RET IN	I/O RETURN In. Control line generated by this card to signal the successful completion of a transaction on the I/O bus. When low, return data is valid for processing by the BIC logic.
-I/O Select	I/O LSA Select. This line is used to enable a this card for I/O operations. When low, this card will respond to the I/O SYNC OUT line provided there is a valid command on the command tag bus.

IBM Internal Use Only

HARDWARE DESCRIPTION

THEORY OF OPERATION

The DIS bus which this card is designed to interface with consists of data, command, handshaking and voltage lines. All signal lines are negative active.

There are eight bidirectional data lines. These lines are gated into a flip-flop on the card at the start of an operation to reduce the loading on the bus to one TTL LS load and also to avoid problems when data is written onto the data bus at the end of the operation. Odd parity for the eight data lines is placed on two unidirectional parity lines; PARITY IN (out of the card) and PARITY OUT (into the card). If PARITY OUT is bad the DAC card ignores the command and does not generate RETURN. The DAC card generates parity across its eight data lines when it generates RETURN.

There are four command lines on the DIS bus. These are input into a decoder which is controlled by the parity checking circuit and the handshaking circuit. When all conditions are met the decoder is enabled and the command processing begins. See "CTC Assignment" on page 9 for command functions.

The handshaking lines are LSA, SYNC and RETURN. LSA selects the card currently addressed by the processor. This is not a bus line, each macrofunction slot has a separate LSA line. A LED is provided on the edge of the card above the Customer Interface Connector which lights when the card is selected. The BIC activates the SYNC line when the data and command tag lines generated by the BIC contain valid data. Both the LSA and the SYNC line must be active for the card to begin operation. When this condition is met the data lines are latched into the flip-flop and the command tag decoding begins. The card activates the RETURN line after the card successfully receives a command and any data being sent to the BIC from the card is valid on the bus. The BIC deactivates the SYNC line after it gets RETURN and latches in the data from the bus. When SYNC is no longer active the card deactivates RETURN and the transaction is complete.

DAC MODULE

The DAC module used on this card is an Analog Devices AD390JD Quad DAC. This hybrid circuit contains four monolithic DAC's in one twenty eight pin module. The module outputs are laser trimmed to a full scale accuracy of $\pm 0.1\%$ with monotonicity guaranteed over the full temperature range. The module output settling time to $\pm 1/2$ LSB is 8 microseconds maximum. Each channel has two twelve bit latches in series. In this application the first latch in each pair is kept in a transparent state and the second is used to latch the data. The DAC outputs are all on a ± 10 volt scale. These signals are translated by the analog output circuitry on the card to give the selected output ranges.

DAC DATA FORMAT

The data written to the DAC is simply an unsigned twelve bit word. Figure 4 on page 8 shows DAC data and the card outputs. Figure 5 on page 8 gives an equation for translating volts to DAC data and the constants needed in the equation for each output range.

IBM Internal Use Only

HEX DATA	OUTPUT VOLTAGE RANGE					
	0 to +5	0 to +10	±2.5	±5	±10	
000	0.0000	0.0000	-2.4988	-4.9976	-9.9951	V
7FF	+2.4994	+4.9988	0.0000	0.0000	0.0000	V
800	+2.5006	+5.0012	+0.0012	+0.0024	+0.0049	V
FFF	+5.0000	+10.0000	+2.5000	+5.0000	+10.0000	V

Figure 4. DAC Card Analog Output vs Digital Input

$$\text{DATA} = \frac{\text{Output in Volts} + \text{OFFSET}}{\text{LSB}}$$

Conversion Formula - Volts to DAC Data

	OUTPUT VOLTAGE RANGE					
	0 to +5	0 to +10	±2.5	±5	±10	
LSB	0.0012210	0.0024420	0.0012207	0.0024414	0.0048828	V
OFFSET	0	0	2.4988	4.9976	9.9951	V

Figure 5. DAC Data Conversion Equation and Constants

OUTPUT RANGE SELECTION

The output voltage range for each channel is selected via the jumpers located below the output calibration pots for the corresponding channel. One of the jumpers determines if the output will be unipolar or bipolar, the other determines the gain of the op amp which follows the DAC module. The available output ranges and their corresponding jumper positions are shown in Figure 6.

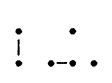
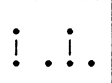
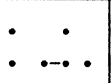
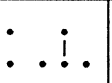
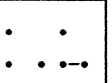
				
0 to +5 volts	0 to +10 volts	±2.5 volts	±5 volts	±10 volts

Figure 6. Jumper Positions and Output Ranges

IBM Internal Use Only

CTC ASSIGNMENT

COMMAND DESCRIPTIONS

CTC 0 addresses the card and turns on the LED to show that the card has been selected.

CTC 1 resets all of the DAC channels and both of the input registers. If the data with this command is x'00' then both input registers are set to x'00' and each of the output channels goes to its most negative value. For example, if a channel is set to the 0-5 volt range then the output of that channel will be 0 volts, or on the ± 10 volt scale the output would be -10 volts. If the data with the reset command is x'FF' then both input registers are set to x'FF' and the output of each channel goes to its most positive value. The reset command does not need to be issued to allow normal card operation. It is provided only for convenience in testing and calibrating the card.

CTC 2,3 write the DAC data to the card. The data from these commands is held in two input buffers, one for the high byte data sent with CTC 2 and one for the low byte and address data sent with CTC 3. The twelve DAC bits are contained in bits 0-7 of CTC 2 data (MSB's) and in CTC 3 bits 0-3 (LSB's). CTC 3 bits 5-7 select the DAC channel which receives the new data. CTC 3 bit 4 is a diagnostic bit. If it is off the data in the input buffers from this CTC 3 and the previous CTC 2 is written to the selected DAC channel. If the diagnostic bit is on the data written to the card is held in the input buffers and is not written to the DAC. This allows the input registers to be tested without affecting the card's outputs.

CTC 6,7 read the contents of the high and low input buffers respectively. The input buffers contain the data most recently written to the DAC or if the diagnostic bit was on in the last CTC 3, the data written for diagnostic purposes.

CTC F returns the unique card ID. The ID of the DAC card is x'1E'.

COMMAND TAG SUMMARY

CTC	FUNCTION	DATA BITS
0	ADDRESS THE CARD	
1	RESET ALL OUTPUTS & REGISTERS	x'00'-most negative x'FF'-most positive
2	WRITE HI BYTE	0-7 - hi byte
3	WRITE LO BYTE	0-3 - DAC LSB's 4 - diagnostic bit 1 - hold data 0 - write to DAC 5-7 - DAC channel
6	READ HI BYTE	0-7 - hi byte
7	READ LO BYTE	0-7 - lo byte
F	READ CARD ID	0-7 - x'1E'

IBM Internal Use Only

COMMAND SEQUENCE EXAMPLES

Normal Operation

- Produce +7.5 volts on channel 2 which is on the 0-10 volt range.

CTC 2 DATA x'A0' Write 8 MSB of data

CTC 3 DATA x'02' Write 4 LSB of data
and address channel 2.

- Produce -2.5 volts on channel 6 which is on the ± 5 volt range.

CTC 2 DATA x'40' Write 8 MSB of data

CTC 3 DATA x'06' Write 4 LSB of data
and address channel 6.

Diagnostic Operation

- Write a test pattern to the card but not the DAC then read it back.

CTC 2 DATA x'AA' Write 'AA' as data

```
CTC 3      DATA x'BB'      Write 'B' as data,
                               the diagnostic bit is on,
                               channel 5 is selected.
```

CTC 6 Data read back should be 'AA'.

CTC 7 Data read back should be 'BB'.

- All channels are either on the 0-5 or 0-10 volt range. Reset them all to zero.

CTC 1 DATA x'00'

IBM Internal Use Only

USER PIN ASSIGNMENTS

CUSTOMER INTERFACE CONNECTOR

The customer interface connector is a 25 pin 'D' shell connector mounted on the edge of the card. This connector has the eight analog signals with a separate shield and return for each channel. Also, a +10 volt reference is included to help in the calibration of this card and the analog input card.

Shielded twisted pair cable is recommended for each channel. The shield on the cable should be connected to the shield line provided for each channel on the DAC customer interface connector and left unconnected at the destination end.

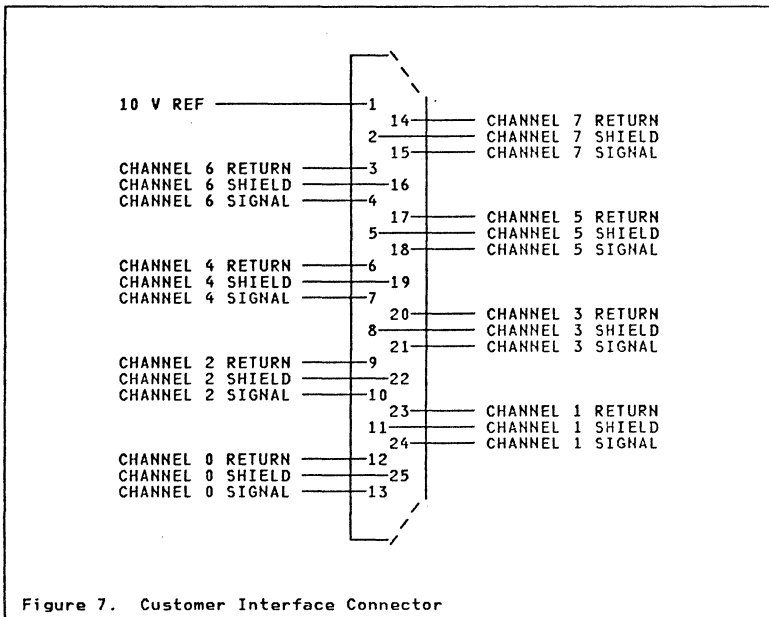


Figure 7. Customer Interface Connector

TEST POINTS

Twenty four test points are provided on the card which allow access to the internal signals on the card. The test points are arranged in two groups, one analog and one digital. The analog pins are located in the upper center of the module side of the card and labeled 'TPA'. There are eight analog signals plus analog ground. These pins are connected in-between the DAC modules and the output circuits. The digital pins are located in the upper left corner on the module side of the card and labeled 'TPD'. They consist of thirteen digital signals plus two digital grounds. They are arranged in two rows with a ground at one end of each row so that the Tektronix 'Harmonica Lead Set' (Tek #012-0800-00 or 012-0968-00) for Tektronix logic analyzers can be used. These lines sample the lines be-

IBM Internal Use Only

tween the control circuit and the data buffers and registers. The test points are shown in the card schematics and in Figure 8 on page 12 and in Figure 9 on page 12

TPD	SIGNAL NAME
1	Digital Ground
2	+Return
3	-Reset (CTC 1)
4	+Load Hi (CTC 2)
5	+Load Lo (CTC 3)
6	-Read Hi (CTC 6)
7	-Read Lo (CTC 7)
8	-Identify (CTC F)
9	-Channel Select 4
10	Digital Ground
11	-Channel Select 3
12	-Channel Select 2
13	-Channel Select 1

Figure 8. Digital Test Points

TPA	SIGNAL NAME
0	DAC 1 Channel 1
1	DAC 1 Channel 2
2	DAC 1 Channel 3
3	DAC 1 Channel 4
4	Analog Ground
5	DAC 2 Channel 1
6	DAC 2 Channel 2
7	DAC 2 Channel 3
8	DAC 2 Channel 4

Figure 9. Analog Test Points

IBM Internal Use Only

CALIBRATION PROCEDURE

Each channel of the DAC card is calibrated independently. Each channel has two pots which must be adjusted; one controls the offset or zero, the other adjusts the gain. On bipolar ranges; adjusting the maximum output, either positive or negative, automatically adjusts the other end of the scale also.

The voltage range for a channel should be selected before that channel is calibrated. If the range of a channel is changed that channel should be recalibrated.

PROCEDURE

1. Set the output to zero: Unipolar ranges - Data x'000'
 Bipolar ranges - Data x'7FF'
2. Adjust the zero pot for that channel (marked 'Z' on the card) until the output is zero volts.
3. Set the output to full scale: DAC Data x'FFF'.
4. Adjust the gain pot for that channel (marked 'G' on the card) until the output reaches full scale for the selected range.

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

Heading ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Heading References</u>
ctcs	DACSPEC	9	CTC Assignment 7

Figure ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Figure References</u>
flow	DACSPEC	3	1:
out	DACSPEC	4	2:
abus	DACSPEC	5	3:
outputs	DACSPEC	8	4: 7
params	DACSPEC	8	5: 7
ranges	DACSPEC	8	6: 8
dshell	DACSPEC	11	7:
tpd	DACSPEC	12	8: 12
tpa	DACSPEC	12	9: 12

Ariel System Memory Card Functional Spec

Control Systems Engineering

**IBM Corporation
General Technology Division
D-035 B-300-47A
East Fishkill
tel 533-7654**

IBM Internal Use Only

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

PREFACE

PURPOSE

This Functional Specification describes the Ariel System MEMORY card. The card's functions, features and requirements will be explained in this document.

RELATED DOCUMENTS

Related documents are:

- Remote DIS User Guide E45-1452 PN 6230664
- Dutchess Technology Users Guide, IBM PN Z45-0053
- Dutchess Fast Circuit Manual, IBM PN Z45-0096
- VTL Application Manual, IBM PN Z45-0414
- Golf Logic Technology Application Manual (MS-438), IBM PN Z45-0855
- Ariel System Memory card Test Specification, IBM PN J000001
- Dover Dynamic RAM, IBM PN 8493494
- Dover Application Specification, IBM PN 6231464

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

TABLE OF CONTENTS

Document Scope	1
Ariel System Memory Card Description Overview	3
Card Functional Overview	3
Capacity Options	3
Logic Functions	3
Performance	3
Interfacing	3
Card Components	4
Card	4
Array	4
Logic	5
Operational Description	7
Addressing	7
Address (SAR) Checking	7
Read And Write Cycles	8
Data Strobing	8
Error Correction Code (ECC) Operations	9
Write Cycles	9
Read Cycles	11
Refresh	13
Refresh Operation	13
Refresh Address Counter	13
Refresh Timer	14
Initialization	14
Card Interface	15
CARD INPUTS	15
CARD OUTPUTS	16
CARD TEST CONNECTOR	16
Card Currents	17
Card Input Signal Voltage Level	17
Data Output Voltage Levels	18
Card Signal Capacitance	18
Card Output Wiring Restrictions	18
Noise Tolerance	18
ESD Requirements	18
Card Jumpers	19
Nucleus Address Protection	19
Power Supply Requirements	21
Power Supply Tolerances	21
Power Supply Noise (at array module)	21
Card Currents and Power	21
Overvoltage/Undervoltage	22
Card Removal	22
Environmental Conditions	23
Module Cooling	23
Mechanical Specifications	23
Timing Application Notes	25
Diagrams	27
Ariel System Memory Card Physical Pins 1-60	28
Ariel System Memory Card Physical Pins 61-120	29
Bill Of Materials	30

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

LIST OF ILLUSTRATIONS

Figure 1.	Ariel System Memory Card Array Addressing and Control Logic	4
Figure 2.	Ariel Memory Card Array Addressing Parity Input Generation	8
Figure 3.	Write Cycle	10
Figure 4.	Read Cycle	12
Figure 5.	Refresh Cycle	13
Figure 6.	Card Layout	31

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

DOCUMENT SCOPE

This document is the engineering specifications for the Ariel System Memory card. The card has been designed for use in the Ariel DSS (Distributed Sensor Subsystem). See related documents listed in the preface of this specification.

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

ARIEL SYSTEM MEMORY CARD DESCRIPTION OVERVIEW

CARD FUNCTIONAL OVERVIEW

Capacity Options

The card contains 1/2 megabytes of dynamic ram plus ECC arranged as 256K x 22. Optionally, sockets are available for 64 kilobytes of either EPROM or EEPROM arranged as 32k x 16.

Logic Functions

- See Figure 1 on page 4.
- Hardware ECC
 - Single bit correct
 - Double error detection
- Array Control and Timing
- Refresh
 - Refresh Address Counter
 - Refresh Timer Request
- Address Multiplexing
- Address Parity Checking
- Write Data Parity Checking
- Read Data Parity Generation

Performance

- Access time (including ECC) = 260 ns
- Read, Write, Refresh cycle time = 330 ns

Interfacing

- VTL, Dutchess Compatible
- Bidirectional Data Input/Output (IO)

IBM Internal Use Only

CARD COMPONENTS

Card

Standard Ariel System Card Dimensions

Array

256K X 22 -22 Dover (256K X 1) modules (see Figure 6 on page 31 for layout).

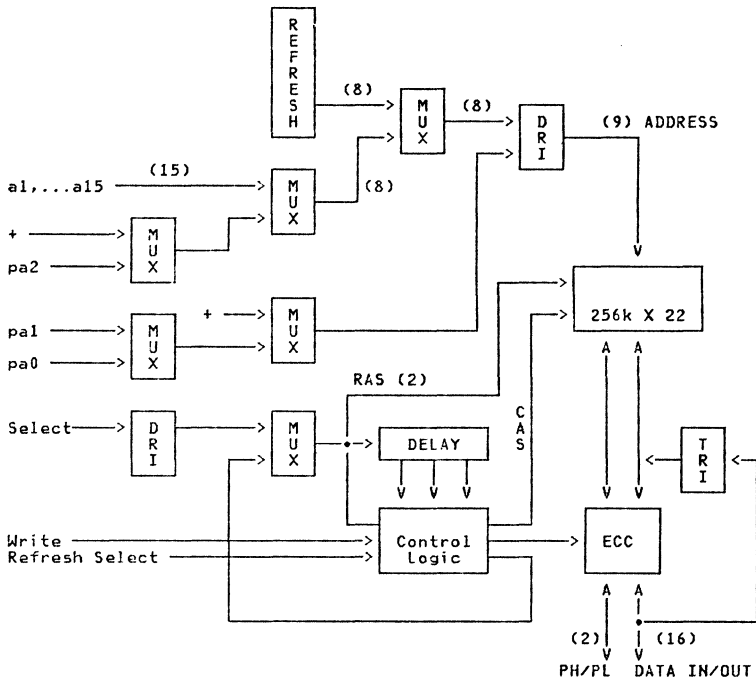


Figure 1. Ariel System Memory Card Array Addressing and Control Logic

IBM Internal Use Only

Logic

- Single bit correction, double error detection, and Write Data Checking - one Golf.
- Refresh Address Counter, Refresh Timer, Address Multiplex and Address Parity Checking - one Golf and an Exclusive-or (74S86).
- Array Control/Timing - one DIF, three Delay lines (50 ns each, tapped every 5 ns), one Quad two to one Multiplexer (74S158), one Octal Inverter Module (74S240), one Dual FF (74LS107), one each of logic chips 74S00, 74S02, 74S05, 74S08, 74S32, and 74S260.
- Array Powering/Redrive - VTL Octal three State Drivers (74S241).
- EProm Control/Timing - one Demultiplexer (74S139) and one 74S00 logic chip.
- EProm Powering/Redrive - two compass drivers.

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

OPERATIONAL DESCRIPTION

ADDRESSING

The addressing into the card is carried by 19 lines. From least to most significance they are:

1. SAR 15
2. SAR 14
3. SAR 13
4. SAR 12
5. SAR 11
6. SAR 10
7. SAR 9
8. SAR 8
9. SAR 7
10. SAR 6
11. SAR 5
12. SAR 4
13. SAR 3
14. SAR 2
15. SAR 1
16. +P0 Sel
17. -CS Pa2
18. -CS Pa1
19. -CS Pa0

From 16 through 19, 32768 word partitions are selected. When +P0 Sel is active, -CS Pa2, -CS Pa1, and -CS Pa0 are all forced inactive. This allows the single line +P0 Sel to select partition 0. Selecting partition 0 (having -CS Pa2, -CS Pa1, and -CS Pa0 all inactive) and not having +P0 Sel active, selects EEPROM memory.

ADDRESS (SAR) CHECKING

The address inputs are low active. The CS ADDRESS PARITY line, together with the 19 address lines above, must have an ODD number of active lines. Array writing is inhibited during parity error conditions.

IBM Internal Use Only

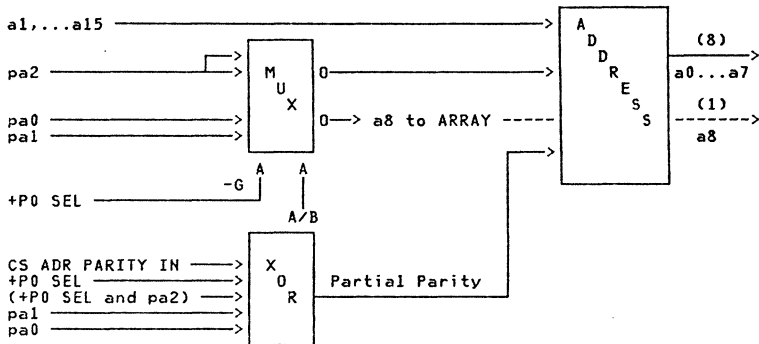


Figure 2. Ariel Memory Card Array Addressing Parity Input Generation: Note: When +P0 SEL is high and pa2 is high, the above MUX will set a16 to high (the MUX not gate output is high). This is address-wise correct for partition 0 select, but is wrong parity-wise. Hence the (p0 SEL and pa2) input reverses parity in this case.

READ AND WRITE CYCLES

A read or write cycle is initiated by activating the Control Store Select signal. All the necessary controls and timing are provided by Ariel System Memory card to complete the storage cycle. Included in these controls are the multiplexing of address bits to the array, array timing control, data strobing through the Error Correction Code (ECC), and checking functions.

Data to be stored or fetched interface with the system via an 18-bit bidirectional data bus. The 18 bits consist of two 8-bit data bytes and an odd parity bit associated with each byte. Two bytes are always stored or fetched, that is, no single byte write capability. To facilitate multiple Ariel System Memory card environments, the data output drivers are open collector.

Data Strobing

External drivers on Ariel System Memory card are controlled with a strobe generated internal to Ariel System Memory card.

IBM Internal Use Only

ERROR CORRECTION CODE (ECC) OPERATIONS

The ECC hardware facility has the following characteristics:

1. Generates ECC check bits based on the incoming 16 data bits coming from the system.
2. Corrects any single bit array failure at the addressed location.
3. Detects 2-bit array failures.
4. Returns 2 data bytes with parity to the system.
5. Provides a 'Disable ECC' capability to assist in Ariel System Memory card testing (VIA the test connector).
6. Checks parity on incoming write data.

Write Cycles

On all write cycles where ECC is enabled the 6 ECC Check bits are automatically generated and stored along with the 16 data bits into the array. The 6 ECC Check bits are generated based on the 16 data bits. The write data parity bits are not included in the check bit generation.

On all write cycles a write data parity check is performed with notification to the system of a check condition on the Data Parity Check line. A write parity check condition inhibits the generation and storing of the data and ECC Check bits.

On write cycles where ECC is disabled, no ECC Check bits are generated. The 22 bits stored into the array are those received from the 18-bit system Data In/Out and 4-bit Test Data In/Out (on the test connector) without modification. Refer to Figure 3 on page 10 for Write Cycle.

IBM Internal Use Only

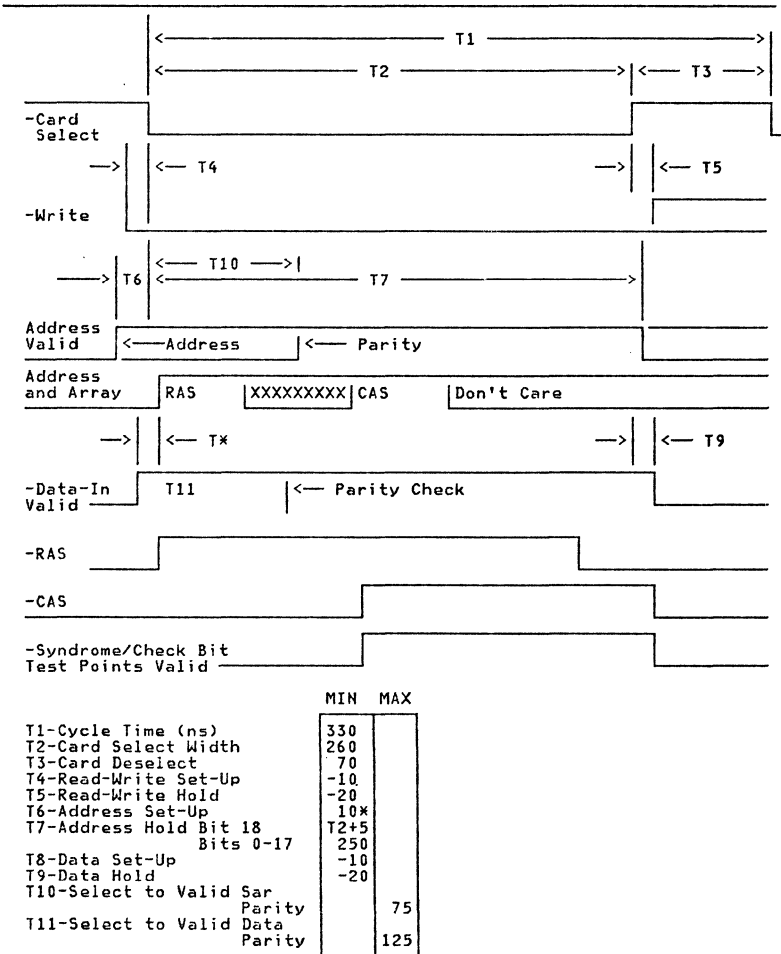


Figure 3. Write Cycle: Note: Falling transition is measured at 0.7 V. Rising transition is measured at 2.4 V.

IBM Internal Use Only

Read Cycles

On all read cycles where ECC is enabled, the 16 data and 6 ECC Check bits received from the array are fed to the error detection/correction logic. Any single bit failure in the 22 stored bits is corrected. The corrected data and parity are returned to the system on the 18-bit data bus. Data parity is generated on the corrected data. Notification to the system of a single bit failure with correction is available on the CS Any Error. All double bit failures are detected, without correction, and notification provided to the system on the CS Uncorrectable error line. Three or more bit failures will yield unpredictable results; for example, failures may not be detected and false corrections may occur.

On read cycles where ECC is disabled, the error detection and correction mechanism is inhibited. The 22 bits received from the array are gated directly to the 18-bit system Data Bus and the 4-bit Test Data Bus, without modification. In ECC disable mode, data from the array is parity checked with system notification on the CS Data Parity Check line. Refer to Figure 4 on page 12 for Read Cycle.

IBM Internal Use Only

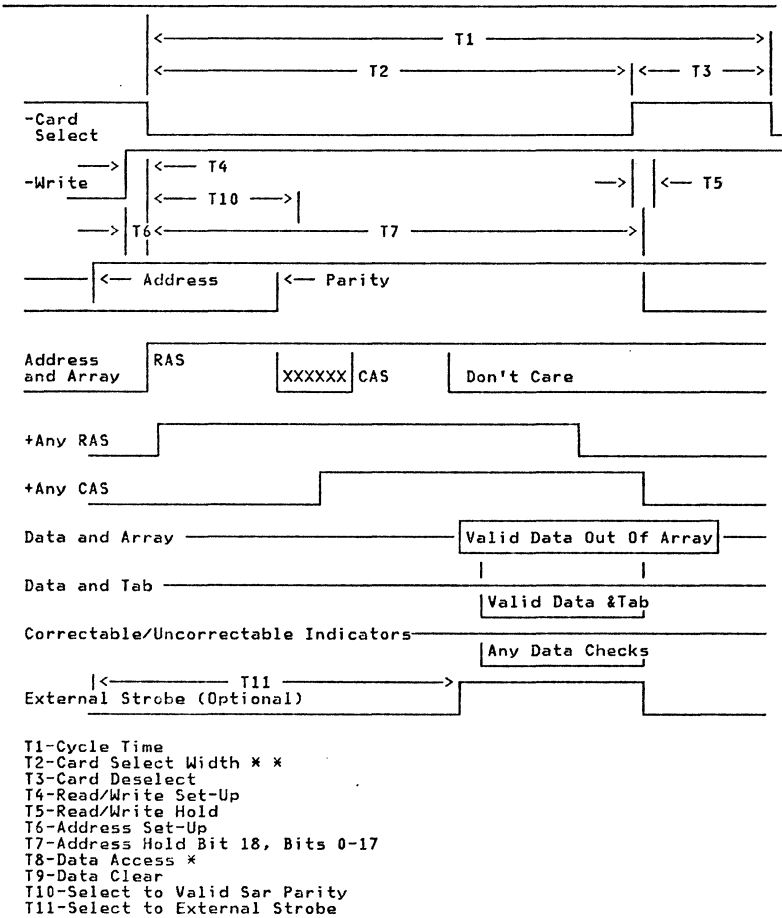


Figure 4. Read Cycle: **If cycle aborted, T2 = 180 ns.

*If External Strobe, Access = T11 + 34 ns. The Row Address must be + Level before normal operation.

Note: Falling transition is measured at 0.7 V. Rising transition is measured at 2.4 V.

IBM Internal Use Only

REFRESH

Since the array is a FET technology, periodic refresh is required to prevent loss of stored data. The entire array must be refreshed within 4 ms. Refresh cycles are required for each of the 256 rows within this period. As a result, one 330 ns Refresh cycle is required nominally every 15.6 μ s. Refer to Figure 5 for Refresh Cycle.

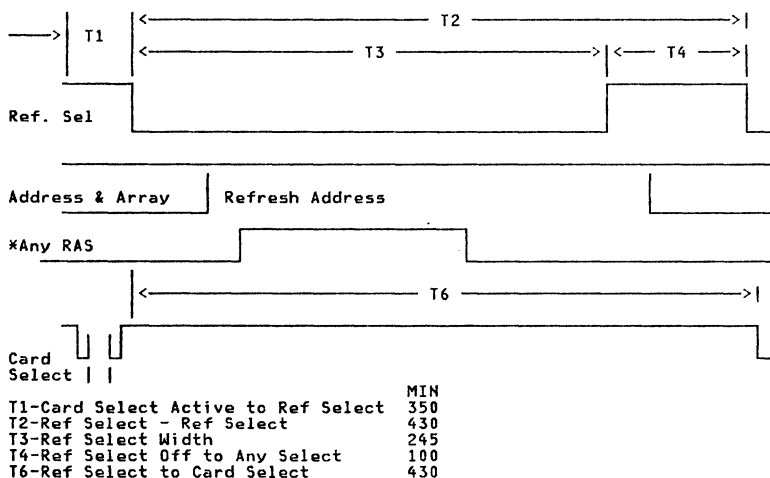


Figure 5. Refresh Cycle: Note; Card Select and Refresh Select Mutually Exclusive.

Refresh Operation

The Refresh operation is controlled by the using system. A Refresh cycle is initiated by the activation of Refresh Select. The Refresh Address and all the necessary controls and timing to complete the refresh cycle are provided by Ariel System Memory card. This includes gating the current value in the Refresh Address Counter to the array modules, performing the array refresh, and incrementing the Refresh Address Counter by one. All modules on the card are refreshed during the refresh cycle. During the time when Refresh Select is active, Control Store Select must not be activated.

Refresh Address Counter

An 8-bit Refresh Address gray code counter is provided on the Ariel System Memory card. The contents of the counter provide the 8-bit refresh address used during a refresh cycle. During a refresh cycle, the counter is incremented by one.

The Refresh Address Counter is reset to zero following a

- Power-On Reset.

IBM Internal Use Only

Refresh Timer

To provide an indication to the system of when a Refresh cycle is required, a Refresh Timer is available on the card. The timer uses the 24 Mhz clock input to the card. Use of the Refresh Timer by the system is optional. When the counter overflows, the Refresh Required signal is activated to the system. No refresh cycle is automatically initiated by the Ariel System Memory card on a counter overflow. It is a system responsibility to activate Refresh Select in response to Refresh Required in a manner consistent with the refresh timing requirements of the array.

INITIALIZATION

One-hundred and twenty eight initialization cycles are required after power-on to ensure proper array operation. This may be accomplished by 128 dummy read, write, or refresh cycles prior to use by the system. One-hundred and twenty eight or more card refresh cycles satisfies this requirement. Once initialization is completed, refresh timing requirement must be met by the system. If power supplies are turned off or refresh requirements are not met, initialization must be repeated before the Ariel System Memory card is ready for system use.

IBM Internal Use Only

CARD INTERFACE

CARD INPUTS

Signal Name	Number of Lines
- CS Select	1
- Refresh Select	1
± SAR 1-15	15
+ P0 Select	1
± Page Select	3
± SAR Parity	1
- P0 Write Enable	1
- EEPROM Program Enable	1
- Write	1
+ Power on Reset	1
+ 24 Mhz Oscillator	1
Total Inputs	----- 27

IBM Internal Use Only

CARD OUTPUTS

Signal Name	Number of Lines
- Data In/Out 0-15 **	16
- Data In/Out PH, PL**	2
- Refresh Required**	1
- CS Data Parity Check	1
- SAR Parity Check	1
- CS Any ECC Error	1
- Any Error	1
+ Inhibit Checks	1
- CS Uncorrectable Data Check	1

Total Outputs	25

**External Pullup Resistor Required

CARD TEST CONNECTOR

Signal Name	Number of Lines
- Test Data In/Out **	4
- Syndrome/Check Bits 0-5*	6
- Read Data Valid*	1
- ECC Disable*	1
- Refresh Timer Check*	1
- Second Refresh Required*	1
- Refresh Address Check*	1
+ Any CAS*	1
+ Any RAS*	1
- CS & EEPROM Select	1
- EEPROM CE	4
- EEPROM WE	1
- EEPROM OE	1

	24

**External Pullup Resistor Required

IBM Internal Use Only

CARD CURRENTS

The following Ariel System Memory card loads occur on the input signals:

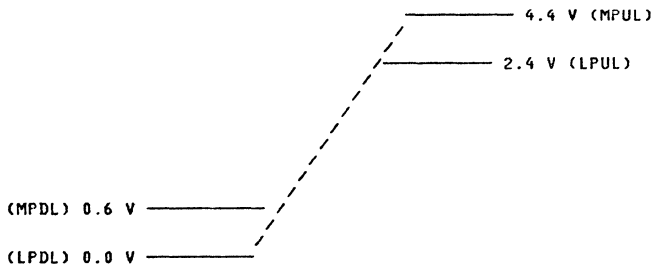
Card Signals Load Current at 0.5 v

Control Store Select (CS)	4.0 mA
Refresh Select	2.8
SAR 0-15	1.8 (+ EEPROMS if inserted)
PA0 Select	6.0
PA1 Select	6.0
PA2 Select	8.0
+P0 Select	10.0
P0 Write Enable	2.0
EEPROM Program Enable	2.0
SAR Parity	2.0
Write	6.0
Power on Reset	2.0
Data In/Out	3.2

The Ariel System Memory card outputs can sink the following currents:
(Maximum)

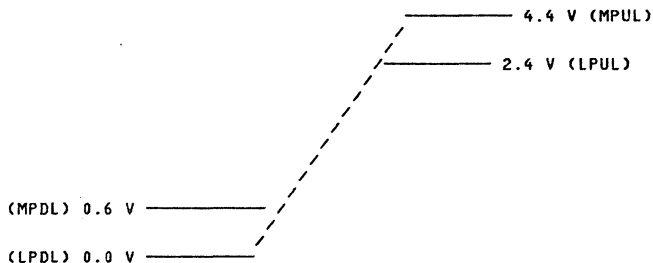
<u>Card Signal</u>	<u>Sink Current at 0.5 V</u>
Data In/Out	14.4 mA
All other outputs	15.6

CARD INPUT SIGNAL VOLTAGE LEVEL



IBM Internal Use Only

DATA OUTPUT VOLTAGE LEVELS



Since the data and status output drivers on Ariel System Memory card are open collector, pullup resistors or resistor dividers must be connected to each data line.

The resistors are not included on the Ariel System Memory card .

Maximum load capacitance ≤ 150 pf.

CARD SIGNAL CAPACITANCE

The Ariel System Memory card capacitance on any input or output signal is ≤ 15 pf.

CARD OUTPUT WIRING RESTRICTIONS

To meet the Golf Driver wiring rules, offcard mainline wire lengths cannot exceed 25 cm. Golf loading rules allow up to eight Ariel System Memory cards to be dotted together.

NOISE TOLERANCE

All data in/out lines drive both Schottky VTL and Golf MS-438 logic. Dutchess Fast Technology is driven by the lines Refresh Select, Power-On Reset, Reference Timer Clock, Inhibit Write, External Strobe, and Enable External Strobe. Card Select, SAR 16, SAR 17, and SAR 18 drive Schottky VTL. All other inputs drive Golf MS-438 logic only. The noise tolerances can be found in the IBM Application Manuals for these technologies.

ESD REQUIREMENTS

Appropriate ESD precautions should be taken.

IBM Internal Use Only

CARD JUMPERS

NUCLEUS ADDRESS PROTECTION

The NUCLEUS partition (partition 0) is write protected through the use of the +P0 WRITE ENABLE line. Card jumpers set the portion of the partition 0 that is actually protected. This allows for the locating of user (communication) tables in partition 0.

JUMPERS	PINS UNUSED	PROTECTED ADDRESS RANGE
Pin2 to Pin3	1	0 to 16K Words
Pin1 to Pin2	3	0 to 24K Words
none	1,2,3	0 to 32K Words

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

POWER SUPPLY REQUIREMENTS

POWER SUPPLY TOLERANCES

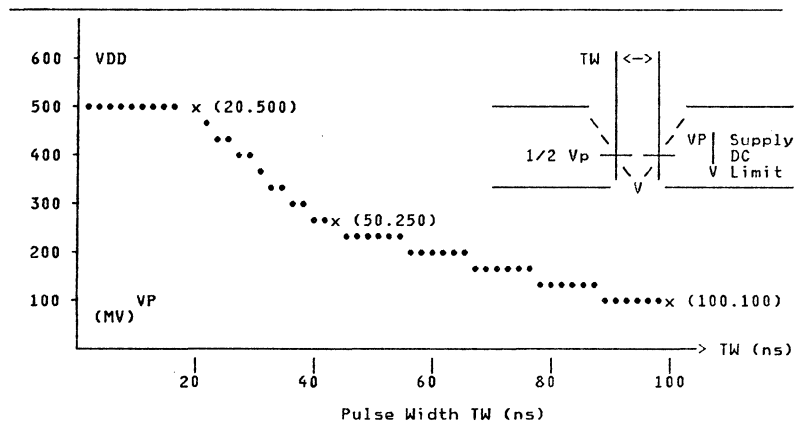
Power Supply Voltages and Tolerances at Card I/O:

+5 V +10%, -5%

POWER SUPPLY NOISE (AT ARRAY MODULE)

Power supply noise shall be limited as follows:

The +5 V average DC value cannot change >0.2 V within a period of <4 ms. The AC noise peak spikes within any cycle cannot exceed 0.5 V peak to peak. These restrictions apply over all operating conditions. The allowable absolute value of positive or negative supply transient voltage as a function of pulse width is as follows:



CARD CURRENTS AND POWER

Capacity	Operating Mode	(Amp) +5 V mA Typical	(Amp) +5 V mA Maximum
512K Bytes	Selected*	$4.25 \times 10^{-7} \times f$	$3.86 \times 10^{-7} \times f$
	Standby (include Refresh)	—	—

IBM Internal Use Only

Capacity	Operating Mode	Watts Typical	Watts Maximum
512K Bytes	Selected*	$_{-} \text{---} + 2.12 \times 10^{-6} \times f$	$_{-} \text{---} + 2.12 \times 10^{-6} \times f$
	Standby (include Refresh)	$_{-} \text{---}$	$_{-} \text{---}$

*f = Average number of card selects per second.

OVERVOLTAGE/UNDERVOLTAGE

Do not permit the 5 V supply to exceed +6.5 V. Remove the voltage from the affected circuits within 40 ms after the voltage exceeds 5.75 V. This is a Golf requirement.

Over Voltage Limit Under Voltage Limit

6.5 V

0 V

CARD REMOVAL

The DC power must be off prior to insertion or removal of the Ariel System Memory card.

IBM Internal Use Only

ENVIRONMENTAL CONDITIONS

The card is designed to operate up to Class C operating environment. The storage temperature range is from -55°C to $+125^{\circ}\text{C}$. See Corporate Specification CES 1-9700-000.

MODULE COOLING

The array module case temperature must not be $>75^{\circ}\text{C}$ or $<10^{\circ}\text{C}$. The Golf and Dutchess module temperature must not exceed 72°C .

MECHANICAL SPECIFICATIONS

In ELSI Packaging, the only stipulation required is that this card is plugged in a board with metal guide posts, PN 811781.

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

TIMING APPLICATION NOTES

1. T0 is defined as the time when the card is selected via Card Select (CS) or Refresh Select.
2. All T-numbers are referenced to T0.
3. The SARS (0-18) may stay at constant up or down levels and must remain constant while card select is active.
4. The card does not latch data in, so all data lines must be held at the proper level in accordance with the timing charts.
5. No significant board skew is assumed on rise and fall time of input signals to card.
6. Input rise and fall times are assumed to be 3 to 5 ns.
7. The SARs 0, 2-8, 17 (row addresses) must stay "+" level before normal operations, if the minimum address set up time is restricted to 10 ns (see :refid=write. and Figure 4 on page 12.)

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

DIAGRAMS

This section contains:

- Physical pin charts
- Bill of materials
- Card outline

IBM Internal Use Only

ARIEL SYSTEM MEMORY CARD PHYSICAL PINS 1-60

<u>IO Pin</u>	<u>Signal</u>	<u>IO Pin</u>	<u>Signal</u>
001		002	
003		004	-TEST PL
005		006	-TEST PH
007		008	+/- DATA BUS 15
009		010	+/- DATA BUS 14
011		012	+/- DATA BUS 13
013		014	+/- DATA BUS 12
015	- REFRESH REQUIRED	016	+/- DATA BUS 11
017		018	+/- DATA BUS 10
019		020	+/- DATA BUS 9
021		022	+/- DATA BUS 8
023		024	+/- DATA BUS 7
025		026	+/- DATA BUS 6
027		028	+/- DATA BUS 5
029	- REFRESH SELECT	030	+/- DATA BUS 4
031		032	+/- DATA BUS 3
033		034	+/- DATA BUS 2
035		036	+/- DATA BUS 1
037		038	+/- DATA BUS 0
039		040	- CS ANY ERROR
041		042	- CS SELECT
043		044	- CS UNCORRECTABLE ERROR
045		046	- CS WRITE
047		048	
049	- ANY ERROR	050	- DATA PARITY CHECK
051		052	- CS ADDRESS CHECK & CS S
053		054	- SAR 15
055		056	- SAR 14
057		058	- SAR 13
059		060	- SAR 12

IBM Internal Use Only

ARIEL SYSTEM MEMORY CARD PHYSICAL PINS 61-120

IO Pin	Signal	IO Pin	Signal
061		062	- SAR 11
063		064	- SAR 10
065		066	- SAR 9
067		068	- SAR 8
069		070	- SAR 7
071		072	- SAR 6
073	- POWER ON RESET	074	- SAR 5
075		076	- SAR 4
077	+ 24 MHZ OSCILATOR	078	- SAR 3
079	+ INHIBIT CHECKS	080	- SAR 2
081	- P0 WRITE ENABLE	082	- SAR 1
083	- P0 WRITE CHECK	084	+ P0 SELECT
085	- PROGRAM ENABLE	086	- CS PA2
087		088	- CS PA1
089		090	- CS PA0
091		092	- CS ADDRESS PARITY
093		094	
095		096	
097		098	
099		100	
101		102	
103		104	
105		106	
107		108	
109		110	
111		112	
113		114	
115	+ 5 VOLTS	116	GROUND
117	+ 5 VOLTS	118	GROUND
119	+ 5 VOLTS	120	GROUND

IBM Internal Use Only

BILL OF MATERIALS

Card Part No. Component Description	Part Number	6912772 256K by 22
Raw Card	6912773	1
ECC Chip Golf	4515514	1
Address Chip Golf	4515529	1
Control Chip-DIF	4515515	1
Compass Driver	6123785	2
Low-End Dover 256K by 1	6231401	22
VTL 74S00 Quad-Nand	2410166	2
VTL 74S02 Quad-Nor	1589220	1
VTL 74S04 Hex-Inverter	8493306	1
VTL 74S05 Hex-Invtr OC	1582505	1
VTL 74S08 Quad-And	1589460	1
VTL 74S32 Quad-Or	1589461	1
VTL 74S86 Exclusive-Or	1589042	1
VTL 74LS107 Dual-JK FF	8493419	1
VTL 74S139 Demultiplexer	1585994	1
VTL 74S158 Multiplexer	1582948	1
VTL 74S240 Tri-State Dr	5615843	1
VTL 74S241 Tri-State Dr	5615844	2
VTL 74S260 Dual-5in Nor	5615409	1
LED	5616982	1
Delay Line	8493466	3
Resistor	2392712	6
Resistor	2408373	1
R-Pac 750/2.3K	2408717	2
R-Pac 390/1.2K	8278619	1
Cap 20 uF	5615372	34
Cap 0.1 uF	1589475	10
Jumper	2396959	1
Socket	72X0327	8

IBM Internal Use Only

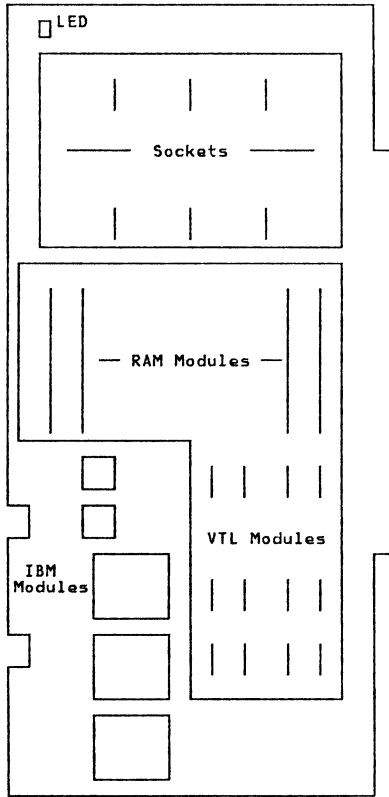


Figure 6. Card Layout

Figure ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Figure References</u>
logic	ROOT	4	1: 3
pargen	ROOT	8	2: 2
write	ROOT	10	3: 9
read	ROOT	12	4: 11, 25
refr	ROOT	13	5: 13
noise	MC	21	6: 6
layout	ROOT	31	6: 4

Imbed Trace

Page iii	PREFAC
Page viii	ROOT
Page 15	MF
Page 17	MA
Page 17	MB
Page 21	MC
Page 27	MD
Page 29	ME
Page 30	OUTLIN

**Distributed Sensor Subsystem:
DIAGNOSTIC/UTILITY
User's Manual**

Control Systems Engineering

**IBM Corporation
General Technology Division
D-035 B-300-47A
East Fishkill
tel 533-7654**

IBM Internal Use Only

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

PREFACE

This document illustrates the use and interpretation of specific diagnostic programs, debug tools, error codes, and utilities available for use with the Distributed Sensor Subsystem (DSS) hardware packages.

REFERENCE DOCUMENTATION

- EDX/J LANGUAGE AND SUPERVISOR DESCRIPTION
- Distributed Sensor Subsystem User Guide (Volume I), Form No.Z45-1123, Part No.6856716
- Distributed Sensor Subsystem User Guide (Volume II), Form No.Z45-1124, Part No.4745244
- Distributed Sensor Subsystem II, Form No.Z45-0925, Part No.6856718
- Distributed Sensor Subsystem III, Form NO.Z45-1081, Part No.6856872
- Distributed Sensor Subsystem IV, Form NO.E45-1289, Part No.7834982
- Jib Prime Functional Specifications, dated August 7, 1978, obtainable through Advanced Chip Development, Department A49, Building 029, General Products Division, San Jose.
- Distributed Interface Card Maintenance ALDs, Form No.Z45-0919, Part No.7866848

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

TABLE OF CONTENTS

Introduction	1
System Initialization	3
Initial Program Load (IPL) Procedures	3
Hazel II, III Bring-up Diagnostics	5
Hazel IV Bring-up Diagnostics	9
Attached Processor/Communication Subsystem Card Bring-up Diagnostics	18
Attached Processor/Disk Adapter Card Bring-up Diagnostics	19
Distributed Interface Initialization	21
Supervisor Utilities	23
Attention Processor	23
Command Syntax	24
\$? DISPLAY SUPERVISOR FUNCTION MENU	25
\$A DISPLAY NAME/LOAD POINT OF ACTIVE PROGRAMS	26
\$C CANCEL PROGRAM	27
\$CP CHANGE PARTITION	28
\$D DISPLAY STORAGE IN HEX	29
\$L LOAD PROGRAM FROM DISK AND ATTACH	30
\$LH LOAD PROGRAM FROM HOST AND ATTACH	30
\$LN LOAD PROGRAM FROM DISK BUT NOT ATTACH	31
\$LNG LOAD PROGRAM FROM HOST BUT NOT ATTACH	31
\$LNUC.. LOAD NUCLEUS FROM DISK	31
\$P PATCH STORAGE IN HEX	32
\$S BEGIN PROGRAM (ATTACH)	33
\$T ENTER DATE AND TIME	34
\$W DISPLAY DATE AND TIME	34
Diagnostic Program	35
Analog to Digital Test (ADTST)	36
Arithmetic Processor Test (APUT)	38
Digital To Analog Utility (DATST)	39
Digital I/O Wrap Test (DIDO)	40
DSC-DSC Communication Test (DSCCOM/DCOMM)	42
Elastic Diaphragm Switch Panel Test (EDSJ)	45
Gaspanel Test (GASPJ)	47
Multi-Function Test (MFTJ)	49
ROC/AUTO-POLLER UTILITY (ROCDG)	52
RS232 Card Test (RS232)	55
Appendix A. Disk Utilities	A-1
Disk Utility Program (DUT)	A-1
Appendix B. DSC Memory Dump Facility	B-1
Memory Dump to S/I Diskette	B-1
DISKETTE INITIALIZATION	B-2
\$DASDI and \$INITDSK	B-2
\$DISKUT1	B-2
DISABLING S/I SENDS TO DSC	B-3
DSC MEMORY DUMP AND DISPLAY	B-3
STARTING DSC TO S/I DUMP	B-3
DUMP PROCEDURE FOR RS232 BASED COMMUNICATIONS	B-4
TERMINATION / RESTART FOR RS232 BASED DUMP	B-4
DUMP PROCEDURE FOR AP/CSS BASED COMMUNICATIONS	B-5
TERMINATION / RESTART FOR AP/CSS BASED DUMP	B-5
DISPLAYING/UPLOADING THE DUMP	B-6
ALLOCATING DUMP DATA SET ON S/370	B-6
FORMATTING THE DUMP ON S/370	B-7
IBM Maintenance Device Attachment to DSS (Standalone Systems)	B-7
DUMP PROCEDURE (MD)	B-8
Appendix C. Debug Utilities	C-1
TEST REQ Key Support	C-1
REGISTER DISPLAY MODE	C-3
DSC II,III Interrupt Level Assignments:	C-3
DSC IV Interrupt Level Assignments:	C-3
MEMORY DISPLAY MODE	C-8
3101 Terminal Debug Support	C-15
HAZEL Debug Tool Overview	C-15

IBM Internal Use Only

Command Entry	C-16
Special Prompt Characters	C-16
Global Commands	C-17
Primary Commands	C-18
Primary Commands Menu	C-18
Display Current CPU's Memory	C-18
Display the Currently Selected CPU's Registers	C-19
Dump Memory to S/1	C-19
Display System Queues	C-19
Return to Previous Primary Subsystem	C-20
Display Global Settings	C-20
Quit Debug	C-21
Stop Jib	C-21
Stop EDX	C-22
Stop Titles	C-23
Redisplay Stop Information	C-23
HAZEL Soft IPL	C-23
Trace EDX	C-23
EDX Trace Size	C-24
View EDX Trace	C-24
Memory Secondary Commands	C-25
Memory Commands Menu	C-25
Patch Memory at Currently Displayed Location	C-25
Patch Memory at Following Last Patch	C-25
Patch Memory at a Specified Address	C-25
Patch Repeat Word in Memory	C-26
Index Through Memory	C-26
Search for Word in Memory	C-26
Transfer Memory	C-26
Scroll through Memory	C-27
Reverse Scrolling Direction	C-27
Register Secondary Commands	C-27
Register Commands Menu	C-27
Select Registers by Chapter and Block	C-27
Display Register Titles	C-28
Patch Local Storage Registers	C-28
Patch HAZEL External Registers	C-28
Patch AP External Registers	C-28
Stop Secondary Commands	C-29
Stop Commands Menu	C-29
Single Step JIB/EDX	C-29
Go to User Specified Address	C-30
Patch EDX Index Registers	C-30
Display Memory	C-30
Appendix D. EPROM Burn Facility	D-1
EPROM Burner Utility (BURN)	D-2
Appendix E. Communication Error Codes	E-1
PROGRAM LOAD ERRORS	E-1
COMMUNICATION SUBSYSTEM RETURN CODES (SEND/RECEIVE)	E-2
DISK ERROR RETURN CODES (NATIVE DISK SUPPORT)	E-2
Appendix F. DIS Error Codes	F-1
Glossary	GL-1

IBM Internal Use Only

LIST OF ILLUSTRATIONS

Figure 1.	Analog To Digital Test display.	37
Figure 2.	DSC-DSC Communication Test Display.	43
Figure 3.	RS232 Test Display.	56
Figure 4.	25 pin 'D' shell connections for RS232 wrap connector.	56
Figure 5.	Disk Utility Option Menu.	A-1
Figure 6.	'RV' (READ VOLUME) function display example.	A-5
Figure 7.	Disk/Diskette Formats and Types.	A-6
Figure 8.	DSC TO S/I DUMP OVERVIEW	B-1
Figure 9.	EPROM burner initial display.	D-3
Figure 10.	EPROM burner utility main option menu.	D-3
Figure 11.	EPROM burner "Run Options" menu.	D-5
Figure 12.	EPROM I card p/n 8912389 data organization.	D-6
Figure 13.	EPROM II card p/n 8692032 data organization.	D-7
Figure 14.	EPROM II card. Locations and data organization for At- tached Processor (AP) EPROM modules.	D-8
Figure 15.	Attached Processor card. EPROM module organization.	D-8

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

INTRODUCTION

The Distributed Sensor Subsystem (DSS) is based on an architecture made up of a microprocessor controller and a set of sensor based I/O function cards. This can be configured to suit about any application in process control or monitoring. The hardware is supplemented by a software operating system (EDX/J), a 'High Level' programming language. This language was written with the Equipment Engineer in mind, containing 'control macros' (DISREADS/WRITES) designed specifically for the control of special I/O function cards. These special I/O function cards are part of what makes up the Distributed Interface System (DIS). A break down of the cards presently available and other information concerning these cards can be found in the DSS User Guide (Volumes I, and II). See 'Reference Documentation'.

The Distributed System Controller (DSC), code name Hazel, makes up the computer controller used to drive the sensor based I/O. There are 3 basic configurations for the Controller.

- Hazel II
- Hazel III
- Hazel IV

The differences between the 3 versions is described in detail in the functional specifications for each controller. Here are a few of the major differences.

The Hazel III is packaged one less card than the Hazel II, plus provides for the installation of an Arithmetic Processor (Intel 8231) module in a socket on the I/O channel card. The Hazel IV provides all the features of the Hazel III plus the following :

- 128k/256k words of random access memory. Where the Hazel II and Hazel III are limited to 64k words.
- Allows the use of a standard video monitor with IBM 3277 Keyboard or the IBM 3101 Display Terminal for the operator console.
- Allows the use of IBM 'TRIM' Floppy Diskette drives and or an optional 10 megabyte 'Hard' file.

The microprocessor which is used to drive the controller is an IBM internal product design, code named 'J1B PRIME'. Detailed information on 'J1B' can be found in J1B Prime Functional Specifications and in the DSS II, III, and IV manuals. See 'Reference Documentation'.

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

SYSTEM INITIALIZATION

This section describes the steps needed to start up or initialize a DSC. For quick reference on how this is done, go to the system initialization example under 'IPL Procedures'. This section is layed out in chronological sequence with respect to the events that take place at initialization time.

INITIAL PROGRAM LOAD (IPL) PROCEDURES

The Distributed System Controller (DSC) is initialized by using one or both of two key switches on the front panel.

DC PWR The DC PWR key switch controls the enable and disable of the DC outputs of the power supply. This switch must be turned to the leftmost position to enable the DC outputs of the supply. To disable the outputs, turn the key switch to the rightmost position. When the key switch is in the 'Enable' position, the key may be removed from the lock.

NOTE : This key switch does not affect the AC power to the supply. AC power on/off is controlled by the Circuit Breaker at the rear of the power supply, or by the main power switch on the AC power distribution box.

Enable Data Buss Out (Enable DBO) Switch

The Enable DBO Switch is responsible for the DSC's 'System Reset'. To perform a system reset, turn this key switch to the rightmost position then turn it back to the leftmost position. This initiates the IPL sequence or 'bootstrap' load. This sequence consists of loading up the DSC memory with the program data needed to run the system. The program data may be loaded in from EPROM, Host via (DSS RS232), Disk, or a combination of these. The data initially loaded into memory will determine how the system will complete the IPL.

If the Hazel IV is equipped with an Attached Processor (AP)/Communication Subsystem (CSS), (AP)/Disk Attachment card set, or both, one of these two card sets will perform the function of the IPL card. The program data will be down loaded from a Series/1 computer ie. (DSS RS232) serial communication interface or from one of the DSS' native disk units.

Board jumpers will determine whether the IPL will be from disk or from Host.

IBM Internal Use Only

System Initialization (Example)

1. Turn on the main circuit breaker found on the Power Distribution Panel. Also verify that all other circuit breakers are ON at the rear of each enclosure making up the DSS.
2. Turn the DC PWR key switch to the enable (leftmost) position. It will take approximately 20 seconds for the Power On Reset (POR) cycle to finish.
3. Turn the 'Enable DBO Switch' to the far right then back to the leftmost position.

If AC and DC power is already up, all that is necessary to IPL is described in step 3.

When the system is initialized as described, it may load all the system software into memory automatically, or it may prompt the operator for a specific program name, of which will be entered via the system console. If the system fails to come up after this procedure has been followed, see the section entitled 'Bring-up Diagnostics' for error codes, descriptions, and possible failure remedies.

IBM Internal Use Only

Hazel II, III Bring-up Diagnostics

The following is a description of the bring-up tests for a DSC II, III with respect to the order in which they are run. The Hex number designated with each of the tests will be displayed on the DIS Data Buss Out (DBO) as each test is run.

Note : This section is in reference to information found in the DSS II, III manuals. See 'Reference Documentation'.

0000 0000 / Hex '00' - Register-Immediate instruction failed

This test checks the CPU's ability to execute each one of the various local storage register-immediate instructions. This test will also fail if a problem arises with one of the local storage registers used during this test.

If this test fails, the card that most likely should be replaced is the 'Engine' or Processor card. This is because the CPU (Jib Prime) and all the local storage registers (Samantha) are physically located on this card.

0000 0001 / Hex '01' - Register-Register instruction failed

This test checks the CPU's ability to execute each one of the various local storage register-register instructions. See test '00' for a possible remedy if a failure occurs.

0000 0010 / Hex '02' - External Register Immediate Instruction failed

This test checks the CPU's ability to execute each one of the various external register-immediate instructions. This test will also fail if a problem arises with the external register interface or with one of the external registers during this test. If this test fails, the following cards may be defective.

- Top card cross over not seated properly on the 'X' connector, or possible bent pins.
- I/O Channel card. Most of the external registers are physically located on this card.
- Clock card (Hazel II). External registers are also located on this card.
- Engine card. External register address and data buss originate from this card.

0000 0011 / Hex '03' - SRL or LDR instruction failed

This test checks the CPU's ability to execute the Shift Right Logical (SRL) and Copy Local Storage Register Even/Odd pairs (LDR) instructions. If a failure occurs, see test '00' for a possible remedy.

IBM Internal Use Only

0000 0100 / Hex '04' - Nucleus longitudinal redundancy code (LRC) failure.
This test verifies that the Data loaded from the IPL card (EPROM) into Hazel's Control Storage (RAM) loaded without error. If a failure occurs, the following cards may be defective.

- Orleans memory card. Most likely would be a card in slot G2 or G4, because these are the cards that the IPL card is loaded into and checked against.
- IPL (EPROM) card. The data on this card may have changed in ROS, or the card may have developed a problem with its control storage or cycle steal interface.
- Engine card. The clock control and part of the control storage interface resides on this card (Hazel III). The Hazel II Engine card contains the control storage interface. It could be interference from a bad tristate driver on the control storage interface that is conflicting with the IPL card.
- Clock card (Hazel II). This card controls the timing for buss switching/multiplexing.

0000 0101 / Hex '05' - Control Store Direct instruction failed
This test checks the CPU's ability to execute each of the Control Store Direct instructions, also referred to as Control Store OP 0 (CS0). This is the first occurrence of any of the Control Store transactions with exception to instruction fetch. If this test fails, the Engine card is most likely at fault because it contains the CPU and the Control Storage interface.

0000 0110 / Hex '06' - Control Store Indirect instruction failed
This test checks the CPU's ability to execute each of the Control Store Indirect instructions, also referred to as Control Store OP 1 (CS1). If this test fails, the Engine card is most likely at fault because it contains the CPU and the Control Storage interface.

0000 0111 / Hex '07' - BALR or Jump instruction failed
This test checks the CPU's ability to execute each of the Branch and Link and Jump instructions. If this test fails the problem is most likely with the Engine card.

0000 1000 / Hex '08' - Local Storage Register Tests failed
This test Ripples zeros through each of the local storage registers and fails if any ones are stuck on. If no stuck ones, then ones are rippled through each of the local storage registers to check for any stuck zeros. If a failure occurs, see test '00' for a possible remedy.

0000 1100 / Hex '0C' - Nucleus/Machine Incompatibility
This test uses the interval timer external registers to tell which type of machine is being used. This is because there are differences in the external register interfaces of the two machines.

If a failure occurs, verify that you are using the correct nucleus for the type of machine being used. If the correct hardware/software combination is being used and a failure persists, there is most likely a problem with the interval timer hardware. This hardware resides on the clock card (Hazel II), or the engine card (Hazel III).

IBM Internal Use Only

0000 1001 / Hex '09' - Interrupt Level Switching Tests failed

This test verifies that the interrupt handling hardware works properly. The test starts out on interrupt priority level 7 and creates a level 6 interrupt then a level 5 interrupt etc. This continues until level 0 is reached then the hardware is forced back to level 7 again. If any one of the interrupt levels didn't switch, the test will fail. Level 7 is the lowest (default) priority, and level 0 is the highest priority (normally used for machine checks). If this test fails, the following cards may be defective.

- Engine card (Hazel III especially). This is because all of the interrupt level switching hardware is on the Hazel III Engine card. The Hazel II Engine card only has the CPU and interrupt level inputs.
- Clock card (Hazel II). This card contains the Interrupt Handler external register module and associated hardware.

0000 1010 / Hex '0A' - Clearing high storage to FFs caused a machine check

This test enables level 0 interrupts (machine checks) then starts to clear memory above the nucleus to FFFFs. This test will fail only if a parity error occurs after level 0 interrupts are enabled. If a failure occurs, the following cards may be defective.

- Engine card. This card contains the interfaces for control storage, local storage, and external registers. Any of these interfaces could be responsible for the parity error.
- Orleans Memory card. There are 8 of these cards. A systematic substitution of a known good card may be necessary to isolate the bad card. It is recommended that before any cards are removed from the board, all cards should be labelled for the card slots they are in.
- IPL (EPROM) card. This card is only used at system reset time, but because it sits on the control storage interface, it may interfere with the address or data buss.

Note : If a parity error was pending from a previous operation this test will fail immediately because this is the first time machine checks are allowed.

IBM Internal Use Only

0000 1011 / Hex '0B' - Memory Tests failed

This test saves the contents of a memory location then stores the address of the location back into that location. The data is read back and compared against the address then the original data is restored. This continues until all the memory locations have been checked. This test will fail if the expected data doesn't match or a parity error occurs. If a failure occurs, one of the following cards may be defective.

- Orleans Memory card. There are 8 of these cards. A systematic substitution of a known good card may be necessary to isolate the bad card. It is recommended that before any cards are removed from the board, all cards should be labelled for the card slots they are in.
- Engine card. This card contains the interfaces for control storage, local storage, and external registers. Any of these interfaces could be responsible for the parity error.
- IPL (EPROM) card. This card only is used at system reset time, but because it sits on the control storage interface, it may interfere with the address or data buss.

Note : This test may fail when using one nucleus but not with another if a parity error is created when the nucleus data is being saved or restored. This is caused by the "word-for-word" differences between nuclei.

IBM Internal Use Only

Hazel IV Bring-up Diagnostics

This group of tests are used to check out the competency of the Hazel IV processor, I/O channel, and Random Access Memory used by the Hazel processor. These tests are run directly after 'POR/IPL', and or after a nucleus is loaded and initialized. The failure codes will be displayed in binary on the DIS Data Buss Out indicators. The "Bring-Up" failure codes are only valid immediately after IPL while the PROCESS_CHK indicator is ON.

NOTE: If the DSC IV is configured with 'Attached Processor' cards, the 'Attached Processor' cards will run their bring-up diagnostics before Hazel is allowed to run its bring-up tests. Thus it may be necessary to check the 'AP' card indicators for any failures if Hazel's DIS DB0 indicators read '1111 1111' / Hex 'FF'. When all of the DIS DB0 indicators are on, this indicates that Hazel may still be held reset by one of the 'AP' cards. Normally, the 'AP' cards release the Hazel reset line when they have finished their initialization, thus allowing Hazel to go through its initialization.

0000 0000 / Hex '00' - Hazel IV instruction tests failed.

This test verifies that each program instruction type can be executed. If this test fails, the Engine card is probably defective. This card contains the CPU and clock control hardware necessary to run this test.

0000 0001 / Hex '01' - Nucleus LRC failed to match that stored.

This test verifies that the data loaded from the IPL card (EPROM) into Hazel's Control Storage (RAM) loaded without error. If a failure occurs, the following cards may be defective.

- Westport Memory Card. There are two of these cards, of which either card could be the problem because their address and data busses are connected.
- Engine card. This card interfaces with the memory address and data buss.
- EPROM card. This card interfaces with the memory address and data buss. Any problem with this interface can cause data errors when loading memory. This card also may have a defective EPROM location thus causing a different LRC to be calculated than the one stored on on this card. This card may have to be replaced with another, paying close attention to getting compatible data burned on the replacement card.

0001 000X / Hex '1X' - Control Storage Instruction tests failed.

0001 0000 - I or D Registers not initially zero after POR.

This test will fail if the I or D registers are not initially zero. If this test fails, the Engine card should be replaced. This card contains all of the control hardware for the I and D register.

0001 0001 - Control Storage Instruction failed.

This test checks out a minimum number of instructions in order to verify that the control storage instructions work properly. The instructions which use the extended addressing functions of the Hazel IV are more extensively checked out in the 'Extended Memory Addressing' test. If this test fails, the Engine card should be replaced. It contains the hardware needed for these functions.

IBM Internal Use Only

0010 000X / Hex '2X' - Local Storage Register tests failed.

0010 0000 - Local Storage Register bits stuck on.

0010 0001 - Local Storage Register bits stuck off.

This test ripples zeros through each of the local storage registers and fails if any ones are stuck on. If no stuck ones, then ones are rippled through each of the local storage registers to check for any stuck zeros. If a failure occurs, the Engine card should be replaced. This card contains all of the local storage register hardware.

0011 0000 / Hex '30' - Nucleus/Machine Incompatibility

This test will fail if a Hazel IV nucleus is loaded into a Hazel III machine. This test uses external register 24 (I reg) to determine whether or not a Hazel IV machine is being used. This register doesn't exist on a Hazel III machine thus making it impossible to write to. If this test fails, verify that you are using the correct nucleus/machine combination. If the proper machine setup is being used, replace the Engine card.

0100 0XXX / Hex '4X' - Interrupt Level Switching tests failed

0100 0000 - SPI to level 6 failed

0100 0001 - SPI to level 5 failed

0100 0010 - SPI to level 4 failed

0100 0011 - SPI to level 3 failed

0100 0100 - SPI to level 2 failed

0100 0101 - SPI to level 1 failed

0100 0110 - SPI to level 0 failed

0100 0111 - PLEX to level 7 failed

This test verifies that the interrupt handling hardware works properly. The test starts out on interrupt priority level 7 and creates a level 6 interrupt then a level 5 interrupt etc. This continues until level 0 is reached then the hardware is forced back to level 7 again. If any one of the interrupt levels didn't switch, the test will fail. Level 7 is the lowest (default) priority, and level 0 is the highest priority (normally used for machine checks). If this test fails, replace the Engine card.

IBM Internal Use Only

0101 0XXX / Hex '5X' - Extended Memory Addressing capability test failed.

0101 0000 - Using (D reg) for (P=1)
0101 0001 - Using (D reg) for (P=2)
0101 0010 - Using (D reg) for (P=3)
0101 0011 - Using (I reg) for (P=1)
0101 0100 - Using (I reg) for (P=2)
0101 0101 - Using (I reg) for (P=3)
0101 0110 - Test data pattern was altered during test.

This test is used to verify that the Hazel processor can successfully move data to/from each memory partition. After each partition operation has been checked, the data pattern used to verify communication is checked for any alterations. If the data pattern has been altered it will be indicated by one of the above failure codes. If this test fails, the Engine card or Westport memory should be replaced.

0110 XXXX / Hex '6X' - Memory Tests failed.

0110 0000 - Address test failed (P=1)
0110 0001 - Address test failed (P=2)
0110 0010 - Address test failed (P=3)
0110 0011 - Inverted Address test failed (P=1)
0110 0100 - Inverted Address test failed (P=2)
0110 0101 - Inverted Address test failed (P=3)
0110 0110 - Parity Inverted test failed (P=1)
0110 0111 - Parity Inverted test failed (P=2)
0110 1000 - Parity Inverted test failed (P=3)
0110 1001 - Address test failed (P=0)
0110 1010 - Inverted Address test failed (P=0)
0110 1011 - Parity Inverted test failed (P=0)
0110 1100 - Clearing Memory to FFFF failed. (P=0)
0110 1101 - Clearing Memory to FFFF failed. (P=1)
0110 1110 - Clearing Memory to FFFF failed. (P=2)
0110 1111 - Clearing Memory to FFFF failed. (P=3)

This test verifies that each memory location can be addressed individually by writing each address with the value of its address. Each address is then tested with the inverse of its address to verify each bit can be changed. The 3rd test done writes each address with the value of its own address, but with one bit inverted to change the parity bit in each location. The 4th test loads each free memory location (memory available to the user) with FFFF. All of these tests are run and checked for each of the 4 memory partitions. Failure codes signify the failing test and the partition being tested.

IBM Internal Use Only

If this test fails, the Engine, Westport memory, Attached Processor (AP), or EPROM (IPL) cards could be defective.

NOTE: The AP and EPROM cards are optional hardware and may or may not be installed in your system. All cards mentioned above connect with the memory interface, thus are capable of producing memory failures. Before replacing any cards, verify that all of the board voltages are within 10% of their ratings. All measurements are to be done on the DSC board.

0111 00XX / Hex '7X' - Timer Interrupt Capability test failed.

0111 0000 - Hot interval timer interrupt

0111 0001 - Hot 1 second timer interrupt

0111 0010 - Wait for interval timer interrupt

0111 0011 - wait for 1 second timer interrupt

This test verifies that both of Hazel's hardware timers run and can interrupt the processor when each timer expires. First, both timers are checked for 'Hot' interrupts by verifying that each timer interrupt can be reset. The test then waits for each timer to expire starting with the interval timer which may take up to 3.2 seconds, then the 1 second timer interrupt is checked. The 1 second timer interrupt should occur while waiting for the interval timer to expire. If this test fails, the Engine card should be replaced.

Note: The APU module on the I/O channel card can cause this test to fail if it has a 'Hot' interrupt, or if no APU is installed, and the jumper on the I/O channel card is not in place. If changing the Engine card doesn't fix the problem, the I/O channel card or the APU module on it should be checked and if necessary, replaced.

IBM Internal Use Only

1000 0XXX / Hex '8X' - External Register tests failed.

1000 0000 - External register (Hex) '06' failed.
1000 0001 - External register (Hex) '07' failed.
1000 0010 - External register (Hex) '08' failed.
1000 0011 - External register (Hex) '0E' failed.
1000 0100 - External register (Hex) '0F' failed.
1000 0101 - External register (Hex) '12' failed.
1000 0110 - External register (Hex) '17' failed.
1000 0111 - External register tests completed.

This test verifies that the External or I/O interface works properly. XR03 is not checked because it is used to display the test failure codes. The registers tested are listed above. If this test fails on any of the registers from (Hex) '06' - '0B', check the top card 'X' crossover connector for any bent pins or misalignment. Then if necessary, replace the I/O channel card. If this test fails on any of the registers from (Hex) '0E' - '17', replace the Engine card.

IBM Internal Use Only

EXTENDED CODES FOR 256K WORD MACHINES

On machines loaded with EDX/J version 5.03 or higher, the remainder of the bring-up test codes will apply. These codes reflect the 8 partition memory feature if installed. Also these tests will state the number of installed memory partitions on the COMMAND or CTC LED indicators during the memory tests. Previously those indicators were always off during the tests.

0101 0XXX / Hex '5X' - Extended Memory Addressing capability test failed.

0101 0000 - Using (D reg) for (P=1)
0101 0001 - Using (D reg) for (P=2)
0101 0010 - Using (D reg) for (P=3)
0101 0011 - Using (D reg) for (P=4)
0101 0100 - Using (D reg) for (P=5)
0101 0101 - Using (D reg) for (P=6)
0101 0110 - Using (D reg) for (P=7)
0101 0111 - Using (I reg) for (P=1)
0101 1000 - Using (I reg) for (P=2)
0101 1001 - Using (I reg) for (P=3)
0101 1010 - Using (I reg) for (P=4)
0101 1011 - Using (I reg) for (P=5)
0101 1100 - Using (I reg) for (P=6)
0101 1101 - Using (I reg) for (P=7)
0101 1110 - Test data pattern was altered during test.

This test is used to verify that the Hazel processor can successfully move data to/from each memory partition. After each partition operation has been checked, the data pattern used to verify communication is checked for any alterations. If the data pattern has been altered it will be indicated by one of the above failure codes. If this test fails, the Engine card or Westport memory should be replaced.

IBM Internal Use Only

HEX '60' to '7F' - MEMORY TESTS FAILED.

0110 0000	- Address test failed	(P=1)
0110 0001	- Address test failed	(P=2)
0110 0010	- Address test failed	(P=3)
0110 0011	- Address test failed	(P=4)
0110 0100	- Address test failed	(P=5)
0110 0101	- Address test failed	(P=6)
0110 0110	- Address test failed	(P=7)
0110 0111	- Inverted Address test failed	(P=1)
0110 1000	- Inverted Address test failed	(P=2)
0110 1001	- Inverted Address test failed	(P=3)
0110 1010	- Inverted Address test failed	(P=4)
0110 1011	- Inverted Address test failed	(P=5)
0110 1100	- Inverted Address test failed	(P=6)
0110 1101	- Inverted Address test failed	(P=7)
0110 1110	- Parity Inverted test failed	(P=1)
0110 1111	- Parity Inverted test failed	(P=2)
0111 0000	- Parity Inverted test failed	(P=3)
0111 0001	- Parity Inverted test failed	(P=4)
0111 0010	- Parity Inverted test failed	(P=5)
0111 0011	- Parity Inverted test failed	(P=6)
0111 0100	- Parity Inverted test failed	(P=7)
0111 0101	- Address test failed	(P=0)
0111 0110	- Inverted Address test failed	(P=0)
0111 0111	- Parity Inverted test failed	(P=0)
0111 1000	- Clearing Memory to FFFF failed.	(P=0)
0111 1001	- Clearing Memory to FFFF failed.	(P=1)
0111 1010	- Clearing Memory to FFFF failed.	(P=2)
0111 1011	- Clearing Memory to FFFF failed.	(P=3)
0111 1100	- Clearing Memory to FFFF failed.	(P=4)
0111 1101	- Clearing Memory to FFFF failed.	(P=5)
0111 1110	- Clearing Memory to FFFF failed.	(P=6)
0111 1111	- Clearing Memory to FFFF failed.	(P=7)

IBM Internal Use Only

This test verifies that each memory location can be addressed individually by writing each address with the value of its address. Each address is then tested with the inverse of its address to verify each bit can be changed. The 3rd test done writes each address with the value of its own address, but with one bit inverted to change the parity bit in each location. The 4th test loads each free memory location (memory available to the user) with FFFF. All of these tests are run and checked for each of the 8 memory partitions. Failure codes signify the failing test and the partition being tested.

If this test fails, the Engine, Westport memory, Attached Processor (AP), or EPROM (IPL) cards could be defective.

NOTE: The AP and EPROM cards are optional hardware and may or may not be installed in your system. All cards mentioned above connect with the memory interface, thus are capable of producing memory failures. Before replacing any cards, verify that all of the board voltages are within 10% of their ratings. All measurements are to be done on the DSC board.

IBM Internal Use Only

1000 00XX / Hex '8X' - Timer Interrupt Capability test failed.

- 1000 0000 - Hot interval timer interrupt
- 1000 0001 - Hot 1 second timer interrupt
- 1000 0010 - Wait for interval timer interrupt
- 1000 0011 - wait for 1 second timer interrupt

This test verifies that both of Hazel's hardware timers run and can interrupt the processor when each timer expires. First, both timers are checked for 'Hot' interrupts by verifying that each timer interrupt can be reset. The test then waits for each timer to expire starting with the interval timer which may take up to 3.2 seconds, then the 1 second timer interrupt is checked. The 1 second timer interrupt should occur while waiting for the interval timer to expire. If this test fails, the Engine card should be replaced.

Note: The APU module on the I/O channel card can cause this test to fail if it has a 'Hot' interrupt, or if no APU is installed, and the jumper on the I/O channel card is not in place. If changing the Engine card doesn't fix the problem, the I/O channel card or the APU module on it should be checked and if necessary, replaced.

1001 0XXX / Hex '9X' - External Register tests failed.

- 1001 0000 - External register (Hex) '06' failed.
- 1001 0001 - External register (Hex) '07' failed.
- 1001 0010 - External register (Hex) '0B' failed.
- 1001 0011 - External register (Hex) '0E' failed.
- 1001 0100 - External register (Hex) '0F' failed.
- 1001 0101 - External register (Hex) '12' failed.
- 1001 0110 - External register (Hex) '17' failed.
- 1001 0111 - External register tests completed.

This test verifies that the External or I/O interface works properly. XR03 is not checked because it is used to display the test failure codes. The registers tested are listed above. If this test fails on any of the registers from (Hex) '06' - '0B', check the top card 'X' crossover connector for any bent pins or misalignment. Then if necessary, replace the I/O channel card. If this test fails on any of the registers from (Hex) '0E' - '17', replace the Engine card.

IBM Internal Use Only

Attached Processor/Communication Subsystem Card Bring-up Diagnostics

This group of tests are used to check out the competency of the Attached Processor card P/N 6229533 driving the Communication Subsystems card P/N 6256814. These cards are features available for use with the DSC IV processor.

The failure codes will be found on the LED indicators located near the top of the AP at the card edge. The most significant bit (MSB) being nearest the top. The following is a list of the codes in use.

11100 - All instructions and LSR space failed.

This test insures that each JIB PRIME instruction can be executed, and tests all of the local storage registers for any stuck bits. If this test fails, the AP card should be replaced.

11010 - RAM Test.

This test verifies the competency of each RAM address, and performs a bit test on each location to check for any stuck bits. The RAM starts at address 4000 (HEX) and ends at BFFF. All of the RAM is located in the SUNSET module located on the the AP card.

If this test fails verify that the card has all the proper voltages, (+5V, -5V, and +8.5V). These voltages must be within 10%. If they are, then the AP card should be replaced.

10110 - External storage register tests.

These tests verify that the external register interface works properly. Only certain external registers are tested because many of the registers are read only, or not program testable.

If a failure occurs replace the AP card.

11011 - Start CSS Channel test pattern.

This pattern only signifies the start of the CSS tests.

1XXX1 - CSS Channel tests.

This group of tests run on each of the 3 channels in the CSS card. The purpose is to check out the operation of the MPCA modules, which make up the communication channel control hardware. These tests don't test the channel EIA drivers and receivers on the CSS card. As the tests run a message is sent to each 3101 display with the status of each CSS channel as it is tested. The LEDs marked in Xs represent the the status of each CSS channel (1,2 & 3) respectively. An LED flashing indicates a failure on that channel.

Channel interrupt failures are indicated by a two word code sent to the 3101 display. The first word is a subroutine return offset from the calling routine. The second word, broken into bytes, the first being the expected interrupt register value, and the second the actual value.

If a failure occurs, verify that the AP/CSS card crossover connector is properly seated on the 'IW' connector. If no problem found, replace the CSS card first, then if necessary, replace the AP.

11000 - Interrupt Level switching tests.

This test verifies that the interrupt switching hardware works properly in the AP. The tests starts on level 0, which is the highest priority level, then switches to level 7 (lowest priority). The test then switches up to level 6, level 5, 4 etc. back up to level 0. This test will fail if it doesn't start on level 0, indicating the card wasn't properly reset after power on, or if any of the levels fail to switch properly. If a failure occurs, the AP card should be replaced.

11111 - Tests Completed

IBM Internal Use Only

Attached Processor/Disk Adapter Card Bring-up Diagnostics

This group of tests are used to check out the competency of the Attached Processor card P/N 6229533 driving the Disk Attachment card P/N 8692434. These cards are features available for use with the DSC IV processor.

The failure codes will be found on the LED indicators located near the top of the AP at the card edge. The most significant bit (MSB) being nearest the top. The following is a list of the codes in use.

11100 - All instructions and LSR space failed.

This test insures that each JIB PRIME instruction can be executed, and tests all of the local storage registers for any stuck bits. If this test fails, the AP card should be replaced.

11010 - RAM Test.

This test verifies the competency of each RAM address, and performs a bit test on each location to check for any stuck bits. The RAM starts at address 4000 (HEX) and ends at 5FFF. All of the RAM is located on the SUNSET module located on the the AP card.

If this test fails verify that the card has all the proper voltages, (+5V, -5V, and +8.5V). These voltages must be within 10%. If they are, then the AP card should be replaced.

10110 - External storage register tests.

These tests verify that the external register interface works properly. Only certain external registers are tested because many of the registers are read only, or not program testable.

If a failure occurs replace the AP card.

11000 - Interrupt Level Switching tests.

This test verifies that the interrupt switching hardware works properly in the AP. The tests starts on level 0, which is the highest priority level, then switches to level 7 (lowest priority). The test then switches up to level 6, level 5, 4 etc. back up to level 0. This test will fail if it doesn't start on level 0, indicating the card wasn't properly reset after power on, or if any of the levels fail to switch properly. If a failure occurs, the AP card should be replaced.

10011 - Disk Adapter Communication Failed

This test verifies that the AP can successfully communicate with the external registers resident on the Disk Adapter card. If this test fails, check the top card 'W' connector for proper connection, then if necessary replace the Disk Adapter card.

10010 - Disk Controller Communication Failed

This test verifies that the AP can successfully communicate with the registers that exist in the MCM disk controller module on the Disk Adapter card. If this test fails, check the top card 'W' connector for proper connection, then if necessary, replace the Disk Adapter card.

IBM Internal Use Only

0XXXX - Test all drives for Index and Ready

This is the last test done in the AP/Disk Adapter card bring-up tests. This test verifies the following :

1. A diskette is properly inserted in the drive.
2. The drive is running at the proper speed.
3. The door on the drive is closed.
4. The proper diskette type is being used.

Before this test is run the LED indicators are turned off. This is only for a very short period of time but if you watch, you can see this happen after the other tests have run. This test is the last to write out to the LED indicators thus leaving initial status of all 4 possible drives on the indicators. The first LED will always be off, and the remaining LEDs (2,3,4, & 5) indicate drive status for drives (1,2,3, & 4) respectively.

- LED 'ON' indicates - Drive NOT ready.
- LED 'OFF' indicates - Drive ready.

Failures : If a failure is indicated on any of the drives you have connected, verify that the cables are attached properly to the drives, the main circuit breaker is turned on for each drive connected, and a diskette (if required) is properly loaded in the drive. If all drives failed, verify that the 'Y' and 'Z' top card crossover connectors are properly seated.

The following is a list of possible failure remedies.

- All drives failed.
 - Replace the disk VFO card. This card is responsible for the interface between the Disk Adapter card and the disk drives.
 - Replace the Disk Adapter card. This card may have an interface problem which was not detected in the bring-up tests.
 - Replace the AP card. This card is responsible for controlling the Disk Adapter card, and transferring the data from the disk drives into the DSC's memory.
- A single drive failed.
 - It will be necessary to determine if the problem is with the drive itself, or with the disk control hardware. This is done by just moving the cables from the bad drive to a working drive and may be done with power on. Afterwards, reIPL and check the AP's indicators again. It is evident that if the good drive fails, that the problem is either in the disk adapter cards, or in the cable attaching them to the drive. On the other hand, when the drives are swapped you may notice that the problem follows a particular drive. Verify the 4 requirements listed previously for the suspected drive, and if they are all met, then replace the drive.

IBM Internal Use Only

DISTRIBUTED INTERFACE INITIALIZATION

The DIS interface goes through an initialization following the 'Bring-up' diagnostics. The EDX/J nucleus interrogates each possible Block Interface Card (BIC) address (0-F) to set up a list containing each hardware configured block of DIS. This list is called the BIC Poll List and is used by the EDX/J Supervisor to service hardware interrupts on the DIS channel. Each BIC found on the system is initialized in the following sequence.

1. Select the BIC. => CTC = 0 (This event is timed and a delay value will be stored in \$BICPLST with the 'Acknowledging' BICs).
2. Read the BIC's interrupt register => CTC = 5
3. Write the BIC's reference register with the inverse value of the interrupt register. => CTC = 3
4. Write the BIC's control register to enable interrupts. => CTC = B, DATA = x'80'
5. Reselect the BIC. => CTC = 0
6. Write the Reference register equal to the Interrupt register. => CTC = 3

When the DIS interface has been fully initialized, the system will be ready for use. At this point the user programs may be loaded. It should be noted that only the BIC is initialized by the EDX/J nucleus. The application programs loaded after the initialization will have to perform any required initialization of the DIS macrofunction cards.

IBM Internal Use Only

THIS PAGE INTENTIONALLY LEFT BLANK

IBM Internal Use Only

SUPERVISOR UTILITIES

This section contains information on the EDX/J Supervisor Utility Functions. The purpose of this section is to illustrate their use only. This is written in reference to the EDX/J LANGUAGE AND SUPERVISOR DESCRIPTION manual.

There are various functions or commands which provide the user with certain privileges outside the application program control. These are called Supervisor Utilities and all are executed from the system console.

These are used online for such tasks as :

- Displaying memory
- Patching memory
- Setting the Time of Day clock
- Loading and cancelling programs.

ATTENTION PROCESSOR

The Attention Processor must be invoked prior to entering any of the Supervisor Utility Commands by pressing the ATTENTION key on the system Console. Depending on which type of system console is being used will determine the ATTENTION key. Note the following :

- When using a standard video monitor and 3277 keyboard for the System Console, press the 'PA1' key.
- When using the IBM 3101 Display Terminal, press the 'PF7' key.

After the Attention Processor has been invoked, you will see a '>_' (greater than) sign followed by the cursor on the display. The desired command and parameters will be entered preceding the cursor. After the command has been typed in, press the 'ENTER' key (3277 keyboard) or 'SEND' key (3101 Display Terminal) to execute the command. The command will be executed and the Attention Processor terminated.

The same procedure is also used when invoking user program attention functions.

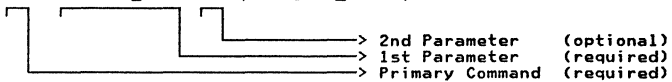
IBM Internal Use Only

COMMAND SYNTAX

Each of the Supervisor Function Utility commands can be entered with all the parameters associated with that command on the same line. A space must be inserted between each of the parameters if this method of key entry is used. This makes a more user friendly interface for individuals that are familiar with the command input format. For individuals who are unfamiliar with the key entry parameters, the basic command may be entered alone and you will be prompted for all other necessary input parameters.

At the beginning of each command the parameters for that command will be listed in the key entry format. The parameter(s) stated in '|' (square brackets) will be required. The other parameters listed will be defaulted to unless specified.

Syntax : "\$LH (program_name key | program_name |)"



Input Example

>\$LH DIAG1 2

This example illustrates the \$LH command being used to load a program (DIAG1) down from the host into partition (2) and attach.

IBM Internal Use Only

\$? DISPLAY SUPERVISOR FUNCTION MENU

Syntax : \$?

The '\$?' command is used to display the supervisor release level and a menu of the valid supervisor utility functions.

Example \$?

- (1) Press the 'ATTENTION' key
- (2) Type in the command '\$?'

>\$?

- (3) Press the 'ENTER' or 'SEND' key

EDX/J4 ASPEN5 09/20/83 SUPERVISOR FUNCTION MENU :

```
$A .... DISPLAY NAME/LOAD POINT OF ACTIVE PROGRAMS
$C .... CANCEL PROGRAM
$CP ... CHANGE PARTITION
$D .... DISPLAY STORAGE IN HEX
$LH ... LOAD PROGRAM FROM HOST AND ATTACH
$LNG .. LOAD PROGRAM FROM HOST BUT NOT ATTACH
$P .... PATCH STORAGE
$S .... BEGIN PROGRAM (ATTACH)
$T .... ENTER DATE AND TIME
$W .... DISPLAY DATE AND TIME
$X .... STOP ADDRESS
```

The above menu may differ slightly from one system to another depending on the supervisor configuration.

IBM Internal Use Only

\$A DISPLAY NAME/LOAD POINT OF ACTIVE PROGRAMS

Syntax : "\$A (key)"

The '\$A' command displays the names and load points of active programs in memory. When performing this command on a Hazel IV machine, all 4 partitions will be displayed unless a partition is specified.

The partition (key) parameter is used exclusively on Hazel IV and should otherwise be excluded.

Example \$A

- (1) Press the 'ATTENTION' key
- (2) Type in the command '\$A'

>\$A

- (3) Press the 'ENTER' or 'SEND' key

PROGRAMS	LOAD POINT	PARTITION
ADTST	5100	0
APUT	8000	1
EDSJ	8000	2
MFTJ	8000	3

>\$A 1 (Press 'ENTER' or 'SEND')

PROGRAMS	LOAD POINT	PARTITION
APUT	8000	1

The program names and load points are displayed by partition.

IBM Internal Use Only

\$C CANCEL PROGRAM

Syntax : "\$C (program_name load_point key | program_name |)"

The '\$C' command is used when there is a need to CANCEL or REMOVE an active program from memory.

The partition (key) parameter is used exclusively on Hazel IV and should otherwise be excluded.

Example \$C

- (1) Press the 'ATTENTION' key
- (2) Type in the command '\$C'

>\$C

- (3) Press the 'ENTER' or 'SEND' key

```
PROGRAM NAME  KEY : MFTJ           (Press 'ENTER' or 'SEND')
MFTJ CANCELLED 09:45:05
```

Equivalent method of key entry :

```
>$C MFTJ           (Press 'ENTER' or 'SEND')
MFTJ CANCELLED 09:45:07
```

```
>$C MFTJ           (Press 'ENTER' or 'SEND')
FOUND FOLLOWING LOAD POINTS WITH KEY :
5600 0000
8000 0001
8000 0003
```

```
ENTER LOAD POINT KEY => 8000 3    (Press 'ENTER' or 'SEND')
MFTJ CANCELLED 10:37:27
```

The last example illustrated the situation when there is more than one occurrence of a particular program name in memory. The individual load points and partitions (keys) of the program name are displayed and you are prompted for the load point and key of the program to be cancelled.

IBM Internal Use Only

\$CP ... CHANGE PARTITION

Syntax : "\$CP (|key|)"

This function is used for changing the active partition (KEY). (Hazel IV ONLY)

Note: This is useful when you don't want to enter the 'KEY' and 'BASE' parameters when displaying or patching memory.

Example \$CP

- (1) Press the 'ATTENTION' key
- (2) Type in the command '\$CP'

>\$CP

(3) Press the 'ENTER' or 'SEND' key

>\$CP KEY : 2 (Press 'ENTER' or 'SEND')

Equivalent method of key entry :

>\$CP 2 (Press 'ENTER' or 'SEND')

Note: Valid partitions which can be entered are 0, 1, 2, and 3.

IBM Internal Use Only

\$D DISPLAY STORAGE IN HEX

Syntax : "\$D (addr word_count key base_addr | addr |)"

The '\$D' function displays the contents of memory online starting at the address specified by the user. Word count, Key, and Base (address offset) are parameters.

The partition (key) parameter is used exclusively on Hazel IV and should otherwise be excluded.

Example \$D

(1) Press the 'ATTENTION' key

(2) Type in the command '\$D'

>\$D

(3) Press the 'ENTER' or 'SEND' key

```
$D : ADDRESS  COUNT  KEY  BASE  ==> 21F8 16 0 0
21F8 C6D6 D3D3 D6E6 C9D5 C740 D3D6 C1C4 407F FOLLOWING LOAD P
2200 D6C9 D5E3 E240 E6C9 E3C8 40D2 C5E8 4040 OINTS WITH KEY
                                $D > A      (to enter new addr)

$D : ADDRESS  COUNT  KEY  BASE  ==> 0 16 0 21F8
21F8 C6D6 D3D3 D6E6 C9D5 C740 D3D6 C1C4 407F FOLLOWING LOAD P
2200 D6C9 D5E3 E240 E6C9 E3C8 40D2 C5E8 4040 OINTS WITH KEY
                                $D > E      (to end display)
```

Equivalent method of key entry :

>\$D 0 16 0 21F8 (Press 'ENTER' or 'SEND')

The above illustrates two ways of displaying the same addresses in memory. The first sets the address to (hex) 21F8 with a word count = 16, key = 0 and base (offset) of 0.

The second sets the address to (hex) 0 with a word count = 16, key = 0, and base (offset) of 21F8.

At the right of the (hex) memory display, the EBCDIC representation of the data is displayed.

Below a method in which the address only was entered. The other parameters default to their present values.

```
>$D 2000 (Press 'ENTER' or 'SEND')
2000 4EF4 F75E FF4F 4E5E 4E40 4A42 7C31 4B0E +47;..!+;+ :.2...
                                $D >
```

Note: After each display you are prompted to enter 'A' for a new address, 'E' to end, or just press enter to display the next sequential area in memory.

IBM Internal Use Only

\$L ... LOAD PROGRAM FROM DISK AND ATTACH

Syntax : "\$L (program_name key | program_name |)"

The '\$L' command is used to load a program from one of the DSS' 'Native' disk units and attach (start). For examples of how this command is used see the '\$LH' command.

The partition (key) parameter is used exclusively on Hazel IV and should otherwise be excluded.

\$LH ... LOAD PROGRAM FROM HOST AND ATTACH

Syntax : "\$LH (program_name key | program_name |)"

The '\$LH' command is used to load a program from the host computer into Hazel's memory and attach (start). The host is connected to Hazel via an RS232c serial communication interface.

The partition (key) parameter is used exclusively on Hazel IV and should otherwise be excluded.

Example \$LH

- (1) Press the ATTENTION key
- (2) Type in the command '\$LH'

>\$LH

- (3) Press the 'ENTER' or 'SEND' key

PROGRAM NAME KEY => MFTJ 2 (Press 'ENTER' or 'SEND')

The program MFTJ will load down from the host and start up in partition 2.

Equivalent method of key entry :

>\$LH MFTJ 2 (Press 'ENTER' or 'SEND')

The following method will load the program into the first available memory location.

>\$LH MFTJ

Note: The program name can be a maximum of 5 characters in length (Host) or 8 characters for Disk. The host contains the 5 character name prefixed by a 3 character communication or tool controller line number assignment.

For example, the program MFTJ might be named T99MFTJ or TXXMFTJ on the host's disk.

TXX => XX = communication line #, or tool controller (TC) line #.

T99 => 99 = program can be accessed by all line numbers

Also if two programs have the same name but one is preceded by a 'T99' prefix, and the other prefixed by the communication or TC line #, the one selected at load time will be the one prefixed by the communication or TC line #.

IBM Internal Use Only

\$LN .. LOAD PROGRAM FROM DISK BUT NOT ATTACH

Syntax : "\$LN (program_name key | program_name |)"

The '\$LN' command is used to load a program from one of the DSS' 'Native' disk units but not attach (start). For examples on the use of this command see the '\$LH' command.

The partition (key) parameter is used exclusively on Hazel IV and should otherwise be excluded.

\$LNG .. LOAD PROGRAM FROM HOST BUT NOT ATTACH

Syntax : "\$LNG (program_name key | program_name |)"

The '\$LNG' command is used to load a program from the host computer into Hazel's memory, but not attach (start).

The partition (key) parameter is used exclusively on Hazel IV and should otherwise be excluded.

Example \$LNG

>\$LNG ==> same as \$LH command but program will not attach !

\$LNUC.. LOAD NUCLEUS FROM DISK

Syntax : "\$LNUC (nucleus_name)"

The '\$LNUC' command is used to load a nucleus from one of the DSS' 'Native' disk units, overlay the present nucleus, and reinitialize the DSC (Hazel).

Example \$LNUC

>\$LNUC TCNUC

The nucleus 'TCNUC' will be loaded and the system will be reinitialized using TCNUC.

WARNING: The use of this command will clear all programs currently loaded in memory and essentially reIPL the system.

IBM Internal Use Only

\$P PATCH STORAGE IN HEX

Syntax : "\$P (addr key base | addr |)"

The '\$P' command is used to alter Hazel's memory online at an address defined by the user.

The partition (key) parameter is used exclusively on Hazel IV and should otherwise be excluded.

Example \$P

- (1) Press the ATTENTION key
- (2) Type in the command '\$P'

>\$P

- (3) Press the 'ENTER' or 'SEND' key

```
$P : ADDRESS KEY BASE ==> 8000 3 0      (Press 'ENTER' or 'SEND')
8000 FFFF FFFF FFFF FFFF FFFF FFFF FFFF .....
> 0 1 2 3 4 5 6 7      (Enter Data then Press 'ENTER' or 'SEND')
                        $P > A      (to enter new addr)
```

```
$P : ADDRESS KEY BASE ==> 8000 3 0
8000 0000 0001 0002 0003 0004 0005 0006 0007 .....
> ,,,,FFFF
                        $P > A
```

```
$P : ADDRESS KEY BASE ==> 8000 3 0
8000 0000 0001 0002 0003 FFFF 0005 0006 0007 .....
>
                        $P > E      (end of patch)
```

Equivalent method of key entry :

>\$P 8000 3 0 (Press 'ENTER' or 'SEND')

The previous examples illustrate two methods of patching memory. The first selects address (hex) 8000, key = 3, and base (offset) = 0. Addresses (hex) 8000 - 8007 were patched, 'A' was entered at the prompt, and the address patched is reentered on the next line for display.

The 2nd example shows a method of patching one or more words in a group by using delimiters for each word to be skipped. The data is entered into word 5 preceded by 4 delimiters. 'A' is entered at the prompt to allow a new address to be entered.

The 3rd example shows the same location in example 2, with word 5 changed from (hex) 0004 to FFFF. The other words remained unchanged. 'E' is entered at the prompt to end the patch utility.

IBM Internal Use Only

\$\$ BEGIN PROGRAM (ATTACH)

Syntax : "\$S (program_name load_point key | program_name |)"

The '\$\$' command is used to attach or start an active program in memory. Programs loaded in using the '\$LNG' command, or from EPROM cards at IPL time are attached using this command.

The partition (key) parameter is used exclusively on Hazel IV and should otherwise be excluded.

Example \$\$

- (1) Press the 'ATTENTION' key
- (2) Type in the command '\$\$'

>\$5

- (3) Press the 'ENTER' or 'SEND' key

PROGRAM NAME KEY : MFTJ (Press 'ENTER' or 'SEND')

Equivalent method of key entry :

>\$S MFTJ (Press 'ENTER' or 'SEND')

FOUND FOLLOWING LOAD POINTS WITH KEY :

5600 0000

8000 0001

8000 0003

ENTER LOAD POINT KEY => 8000 3 (Press 'ENTER' or 'SEND')

The last example illustrated the situation when there is more than one occurrence of a particular program name in memory. The individual load points and partitions (keys) of the program name are displayed and you are prompted for the load point and key of the program to be attached.

IBM Internal Use Only

\$T ENTER DATE AND TIME

Syntax : \$T

The '\$T' command is used to set the date and time in Hazel.

Example \$T

- (1) Press the 'ATTENTION' key
- (2) Type in the command '\$T'

>\$T

- (3) Press the 'ENTER' or 'SEND' key

ENTER DATE MM DD YY : 12 13 83
ENTER TIME HH MM SS : 10 30 15

(Press 'ENTER' or 'SEND')
(Press 'ENTER' or 'SEND')

The date and time entered comes into affect immediately upon pressing ENTER.

\$W DISPLAY DATE AND TIME

Syntax : \$W

The '\$W' command is used to display the current date and time in Hazel.

Example \$W

- (1) Press the 'ATTENTION' key
- (2) Type in the command '\$W'

>\$W

- (3) Press the 'ENTER' or 'SEND' key

DATE : 12/13/83 TIME : 10:30:15

IBM Internal Use Only

DIAGNOSTIC PROGRAM

This section describes the use and interpretation of selected diagnostic programs which can be universally used on most DSS hardware configurations.

The diagnostic programs described in this section should only be used by Certified technicians. These programs give the user absolute control of the DIS hardware. ALL safety interlocks MUST be implemented in hardware--asynchronous to all system programming.

LOADING THE DIAGNOSTIC PROGRAMS: In order to load one or more of the diagnostic programs, you will need the following.

1. You will need to be able to enter data and system commands from the system console (3277 Keyboard W/Video Display or 3101 Terminal) or by a method provided by the system programmer which specifically allows program loading.
2. A Host port (RS232 link to a Series/1) connected, an attached disk drive, or a diagnostic EPROM card set.

The type of system being used will determine the proper command for loading in the diagnostic program(s). There are several commands which can be used and are found in the section entitled 'Supervisor Utilities'.

IBM Internal Use Only

ANALOG TO DIGITAL TEST (ADTST)

This test provides the ability to read all 16 'analog in' channels from the DSS macrofunctions 'Analog to Digital IV' card P/N 8912432 and 'High Accuracy Analog to Digital Converter' card P/N 9402053. The 'Programmable Gain A to D' card P/N 6290670 is also supported in REL 2.1. When this program is loaded, the following will be displayed.

```
*****
*               ANALOG TO DIGITAL CARD TEST               *
* ***** REL 2.0 - mm/dd/yy.hh.mm *****                *
ATTENTION FUNCTIONS:  EN = END TEST
```

THE PRESENT A/D ADDR = 00xx
DO YOU WISH TO CHANGE ADDR OF A/D ? :

If the card address is correct for your application, just hit enter and the 16 multiplexed channels will be displayed to the screen. Else enter 'Y' and you will be prompted for the new card address.

```
ENTER NEW A/D ADDR. (B + M) IN (HEX) = xx
      |
      |----> Macro Addr (1 - F)
      |----> Block Addr (0 - F)
```

EN: The 'EN' attention function is used to end/restart the test. This may be used if you wish to enter a new card address. For more information on the use of attention functions, refer to the section "Attention Processor".

IBM Internal Use Only

Once the desired card address has been entered, all 16 analog channels will be displayed. The following is an example of this display.

```
*****
*          ANALOG TO DIGITAL CARD TEST          *
***** REL 2.0 - mm/dd/yy-hh.mm *****
ATTENTION FUNCTIONS:  EN = END TEST
```

CHANNEL	MILLIVOLTS	CHANNEL	MILLIVOLTS
0	14.64	8	.00
1	14.64	9	-9.76
2	43.92	10	-9.76
3	9.76	11	-9.76
4	9.76	12	.00
5	9.76	13	-4.88
6	9.76	14	-4.88
7	4.88	15	-19.52

Figure 1. Analog To Digital Test display.

IBM Internal Use Only

ARITHMETIC PROCESSOR TEST (APUT)

This test is a general purpose Distributed System Controller (DSC) diagnostic program. The functions provided in this test allow the user to test the competency of the INTEL 8231A Arithmetic Processing Unit (APU) attachment to the DSC II, III, or IV. This is installed in a 'two wide' card option available for DSC II processors P/N 8912398. For the DSC III, and IV processors, a socket is provided on the I/O channel card and a jumper. The jumper should be removed when an APU is installed in the socket.

When this test is loaded, the following menu will be displayed.

***** APU TESTS *****

SELECT APU OPERATION TO BE PERFORMED:

- | | |
|-------------------------------|-----------------------------|
| 1 - SINGLE PRECISION MULTIPLY | 2 - SINGLE PRECISION DIVIDE |
| 3 - DOUBLE PRECISION MULTIPLY | 4 - DOUBLE PRECISION DIVIDE |
| 5 - FLOATING POINT ADD | 6 - FLOATING POINT SUBTRACT |
| 7 - FLOATING POINT MULTIPLY | 8 - FLOATING POINT DIVIDE |

- 9 - TERMINATE PROGRAM
0 - ALL OPCODES

This test uses predetermined values for all of the operations. This makes it possible for this test to run through all of the options in the menu automatically.

0 - ALL OPCODES: This option will run through all of the options listed in the menu automatically 'n' number of times. 'n' can be a value 1-30000 and is prompted for before the test begins.

Note: If the DSC 'PROCESS CHECKS' when this test is run it is because the APU hardware didn't respond in the correct amount of time, 'Stalling' the EDX/J operating system, or a parity error occurred during the test. In either case, the APU, or the card with the APU installed on it, should be replaced.

CAUTION: The APU module is static sensitive. When handling this module it must be done in conformance with the procedures for handling static sensitive parts. Otherwise you may replace a bad APU module with another bad APU.

This test will display any errors encountered during calculations to the system console.

IBM Internal Use Only

DIGITAL TO ANALOG UTILITY (DATST)

This program/utility is used to selectively set the output values of the Multi-Level DAC macrofunction card.

When loaded, the current DAC card address will be displayed, and the user is provided the opportunity to change this value. If the LSA entered is not a Multi-Level DAC card, it will be asked that the address be reentered.

THE PRESENT DAC ADDR = 00xx
DO YOU WISH TO CHANGE ADDR ? : Y
ENTER NEW ADDR. (B + M) IN (HEX) = 04 <= block 0 macro 4 (example)

Once the correct LSA has been entered, the following will be displayed:

ANALOG OUT (DAC) UTILITY ATTN EN TO END
BLOCK MACRO 0004
OUTPUT CHANNEL 0
MIL. VOLTS 0
DAC DATA = 0000

ENTER OUTPUT CHANNEL 0 - 7 (0) <= channel 0 entered
0

After selecting the desired channel, a prompt for the output value will be provided.

ANALOG OUT (DAC) UTILITY ATTN EN TO END
BLOCK MACRO 0004
OUTPUT CHANNEL 0
MIL. VOLTS 0
DAC DATA = 0000

ENTER MIL. VOLTS 0 - 5000(0) <= output value entered
2500

After entering this value, the program will return and prompt the user for the next output channel value to be written. To end this utility or enter a new card address the "EN" attention function is used.

IBM Internal Use Only

DIGITAL I/O WRAP TEST (DIDO)

This test provides the capability of writing data to a specified digital output group and reading that data back on a specified digital input group. The supported digital I/O cards are listed later in this section.

Note: This test requires that "symmetrical wiring"¹ is used from the card I/O to the cable edge connector. This insures that proper connection will be made when the wrap cables are in place. The wrap cable is terminated on each end with a male 25-pin "D" shell connector. The cable is configured pin-for-pin.

When attached the program will display the following:

```
----- DI/DO DIAGNOSTIC -----  
---- R 2.0          mm/dd/yy-hh.mm -----  
>EN = QUIT -----  
  
ENTER DO LSA IN HEX XX  xx  
ENTER DI LSA IN HEX XX  xx  
                        |  
                        |> macro_addr (0-F)  
                        |> block_addr (0-F)
```

The cards selected for the wrap test must be of compatible types. These types are separated into two (2) groups:

1. Programmable DI/DO card type, or
2. nonprogrammable DI/DO type.

The nonprogrammable types include:

- 16 Digital Output/Digital Input cards.
- EIA Digital Input card
- Digital Output II card

Before entering the card LSAs, connect the wrap cable to the respective cable connectors. The wrap cable must be connected first because the test will start as soon as the last LSA is entered. If the wrap test passes, a message will be displayed stating "Test Completed", else a message similar to the following will be displayed.

```
WROTE  FFFF 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
READ   0000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

This is an example of the situation where the wrap cable wasn't connected, or, it was connected incorrectly.

All of the examples thus far, have been for the "nonprogrammable" digital card types. If the digital output LSA entered happens to be a Programmable DI/DO card, the information entered will be done in a slightly different format.

48 BIT DI DO TEST
ARE CABLES IN PLACE ? Y

¹ Inputs and outputs have identical cable : n-outs.

IBM Internal Use Only

```

ENTER DO GROUP NUMBER
1 = 0 - 7
2 = 8 - 15
3 = 16 - 23
4 = 24 - 31
5 = 32 - 39
6 = 40 - 47
1
ENTER DI GROUP NUMBER
3
ERROR      EXPECTED BITS  8000 1 0 0 0  0 0 0 0
              READ BITS   0000 0 0 0 0  0 0 0 0
CONTROL REG = 003C
DO CTC = 0F02      DI CTC = 0F0A
  
```

This example shows a bit failure when trying to wrap groups 1 and 3.

Note: The group numbers selected must not be within the same "16 bit" range or a warning message and a request for the groups to be reentered will be displayed. This is only a software dependency, the hardware supports 8-bit groups. The "16 bit" ranges are :

Group 16 Bit Range

```

1,2      Range 1
3,4      Range 2
5,6      Range 3
  
```

Thus when selecting groups to test, selecting group 1 and 3 for the DO and DI group will work, where selecting group 1 and 2 won't be allowed. Groups 2 and 3, however, will be allowed as the DO and DI groups.

EN: The attention function "EN" when entered will cause a "break" at the first available time thus allowing the test to be restarted or ended. To use, press the "Attention Key" and enter "EN". Continue to run the test normally. When the test reaches the first break point, the opportunity will be given to restart or end the test.

IBM Internal Use Only

DSC-DSC COMMUNICATION TEST (DSCCOM/DCOMM)

This test is a general purpose Distributed System Controller (DSC) diagnostic program. The test program provides a method of testing the DSC "HOST" communication port under normal system operating conditions.

The EDX/J Send/Receive support is used for all data transfers. This makes possible the use of this program for testing any communication device supported on the "HOST" channel. The two currently supported are:

RS232 (DIS)

A DIS two card set which provides RS232 asynchronous communication support.

CSS Communication Subsystem support. A DSC IV feature card set currently programmed for the RS232 communication protocol.

Theory of Operation

The purpose of this program is to provide a communication "WRAP TEST" which not only checks the logic card integrity, but also verifies that the communication cable connections are correct. In order to provide these functions, a "Destination" system is needed. The "Source" sends data of various length and content to a "Destination" and awaits a mirror image of the send. The Destination's responsibility is to send all data received, back to the originator. This process forms the "WRAP" test. Also since a standard communication connection is used instead of a wrap connector, the interface connections are verified. Each time the wrap is completed, the EDX/J Supervisor's communication statistics are displayed. The statistics show a summary of all retries, discarded messages, desperate attempts to establish communication (BLASTS), and a bad count since IPL time. A total of all unrecoverable errors are also displayed.

IBM Internal Use Only

Set Up Procedures

- Two DSC systems are needed. One will be designated as the "Receiver" or "Destination". The other will be left as the "Source" ("Source" mode is the default).
- Connect a standard host cable between the host ports on the two systems. The communication specifics of these systems must be identical i.e. baud rate, parity, LRC/CRC, stop bits, etc.
- Load the program DSCCOM or DCOMM in both systems. This will probably have to be done by some means other than the host port i.e. Disk or EPROM IPL card. If the host port is functional, then it may be loaded into the DSC via this port, but before the cable is connected in the test mode.
- Select R (Receive Mode) on the system chosen to be the destination. This will immediately blank the display and wait for data from the Source. Communication statistics may also be displayed at the destination by using the P attention command.

Note: The destination statistics are only displayed after data has been received and re-sent back to the source. The P attention command is used to ENABLE/DISABLE the display of the statistics only.

- Press ENTER or SEND at the Source to start the test. Once started the test will run continuously until an unrecoverable error occurs, or, it is ended using the E attention command.

Error Codes

All Error codes are posted by the EDX/J communication support and may be found in this manual. See "Communication Error Codes". All "Hard" errors flagged by this test program are also supplemented with a short word description of the error. This is necessary because many times, a send and receive error will be flagged. The description displayed will indicate which error occurred first. The first error in most cases will be the important one.

```
*****mm/dd/yy-hh.mm*****
*   D S C   C O M M U N I C A T I O N   T E S T   *
*****
ATTENTION FUNCTIONS: E => END TEST  P => DISPLAY OFF/ERR LOOP

STATISTICS:  #RETRY  #DCARD  #BLAST  #BADCHT
              0       0       0       0

SEND ERROR COUNT      : 0      RECEIVE ERROR COUNT      : 0
S/R COUNT MISMATCHES : 0      WORD COMPARE ERRORS      : 0
TOTAL 'HARD' ERRORS   : 1
WORD COUNT            : 1      TEST DATA                : AAAA
=====
```

Figure 2. DSC-DSC Communication Test Display.

The previous figure is an example of the "Source" DSC display while the test is running. All "Hard" errors will be shown below this display with an identification of the error.

The "WORD COUNT" and "TEST DATA" will be varied as the test runs. Each "Test Pattern" will be sent in separate messages for each of the programmed word counts. Also, destination, message length, and check code information is sent with each data pattern. This information is used by the EDX/J Send/Receive support and is not readily available to the user.

User Notes

All test messages are verified for their original length. Any differences in message length between the original source transmission and that received from the destination is flagged as a Count Mismatch.

IBM Internal Use Only

A 100 percent data compare is done on all transactions deemed good by the Send/Receive communication support. If any discrepancies are detected, the condition is flagged as a Data Compare Error.

The "Communication Statistics" shown are actual values stored within the EDX/J Supervisor. The program uses "Supervisor Equates" at compile time to adjust the pointer values used to access this information. Therefore, this program must be compiled at the EDX/J Supervisor level presently being used to insure the integrity of all information displayed.

IBM Internal Use Only

ELASTIC DIAPHRAM SWITCH PANEL TEST (EDSJ)

This program provides a method of exercising an EDS Panel. All of the panel switches can be tested as well as the LED back lights for each of the switches. This test also writes data to the HEX displays.

When this program is loaded, the following will be displayed:

```
START EDS PANEL TEST   mm/dd/yy
****  ATTN FUNCTIONS  ****
IM - INT. MASK CHANGE
EN - END TEST
```

```
BLOCK + MACRO ADDR. = 0F00
WANT TO CHANGE ADDR. ? _
```

If the card address displayed is incorrect, enter 'Y'. You will be prompted for the new card address. After you have entered the desired card address you must enter the correct interrupt bit by changing the interrupt mask. If the proper mask isn't set, you will not be able to test the panel because this program polls for the interrupt(s) stated in the mask register.

IM: The 'IM' attention function is used to change the interrupt mask value. The mask register is 'ANDed' with the BIC interrupt register. If the interrupt bit matches one of the bits in the mask, the Control panel card is then serviced, otherwise the program continues to poll for the interrupt. For information on the use of attention functions, see the section on the 'Attention Processor'.

EN: The 'EN' attention function is used to restart or end the EDS panel program.

IBM Internal Use Only

While the test is running the following example may be displayed:

```
EDS TEST
HEX DISPLAY DATA = 0011
(ATTN IM) INTERRUPT BITS = 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
ROW COLUMN ADDRESS = 0000
OLD COL STATUS = 1 1 1 1 1 1 1
SWITCH BIT = 1 0 0 0 0 0 0
NEW COL STATUS = 0 1 1 1 1 1 1
*****
```

INTERRUPT BITS: The interrupt bits displayed are the content of the BIC interrupt register. When the test is first initialized, make note of the state of these bits, then press one of the EDS panel switches. You should then make note of which of the bits changed state. That will be the interrupt bit the Control Panel card is connected to, and should be entered into the programs interrupt mask for proper program operation. This program defaults the interrupt mask to a x'0080' value (interrupt bit 8).

ROW COLUMN ADDRESS: This value represents the ROW/COL/0/0 coordinate of the last switch pressed.

NEW COL STATUS: This group of bits represent the current state of the LED back lights in the column of the last pressed switch (1, LED = ON).

OLD COL STATUS: This group of bits represent the previous state of the LED back lights before the last pressed switch (1, LED = ON).

SWITCH BIT: This group of bits represent the row position of the last switch pressed.

IBM Internal Use Only

GASPANEL TEST (GASPJ)

This test is a general purpose Distributed Interface diagnostic program specifically used to write data to the Gaspanel assembly P/N 7832973 using GP ROS modules P/N 4943070 and 4944404. The gaspanel communication is done through a standard 16 Digital Output macrofunction card P/N 9402095. When this test is loaded the following will be displayed.

START GASPJ mm/dd/yy
GASP BLOCK + MACRO = 0007 WANT TO CHANGE ? _

If the Digital Output (DO) card to be used for the gaspanel isn't physically located in the above stated board location enter 'Y' and you will be prompted to enter the new card address.

ENTER NEW BLOCK + MACRO IN HEX bm
 ||
 '----> Macro address (1 - F)
 '----> Block address (0 - F)

When you have entered the desired card location the following menu will be displayed.

ENTER : 1 DISPLAY KEYED MESSAGE
 2 WRITE FULL SCREEN
 8 TO CHANGE BLOCK + MACRO
 9 TO END GASPJ
ATTENLIST R = RESTART DI = DIAG DISPLAY

1 DISPLAY KEYED MESSAGE: When this option is selected any data entered at the system console will be displayed to the gaspanel. To return from this option you must press the console's 'Attention' key (see attention processor for info), press enter, then enter R to restart the diagnostic.

By using the 'DI' Diagnostic Display attention function, the EBCDIC representation and the message length will be displayed as you enter each message. To disable the diagnostic display, just repeat the procedure you used to enable the diagnostic display.

2 WRITE FULL SCREEN: This option allows the operator to enter any 'hex' value that the gaspanel can decode and it will be displayed over the entire gaspanel display area. To return from this option, you must use the 'R' attention function.

8 TO CHANGE BLOCK + MACRO: Select this option when you want to enter a new address for the Digital Output (DO) card used to control the gaspanel.

9 TO END GASPJ: This will detach the test. It may be reattached by using the '\$S' attention function if desired.

IBM Internal Use Only

Character Codes: The following is a chart of some of the Hex character codes needed when using option 2 (WRITE FULL SCREEN).

CHARACTER / HEX CODE		CHARACTER / HEX CODE		CHARACTER / HEX CODE	
A	41	P	57	1	71
B	42	Q	58	2	72
C	43	R	59	3	73
D	44	/	61	4	74
E	45	S	62	5	75
F	46	T	63	6	76
G	47	U	64	7	77
H	48	V	65	8	78
I	49	W	66	9	79
J	51	X	67	:	7A
K	52	Y	68	#	7B
L	53	Z	69		
M	54	\	6A		
N	55	,	6B		
O	56	0	70		

IBM Internal Use Only

MULTI-FUNCTION TEST (MFTJ)

This test is a general purpose Distributed Interface diagnostic program. The functions provided in this test allow the user to communicate with the DIS macrofunction cards using the DSC instead of a manual approach using the maintenance panel. This test also gives the ability to prove out on an individual card basis, the state of selected cards using similar resources as the tool application program.

When this test is loaded into the system, the following menu will be displayed.

```
HAZEL MULTI FUNCTION TEST  VERS. 1.0 mm/dd/yr
ENTER  1  TO READ ALL ID'S
        2  TO DO A SINGLE READ
        3  TO DO A SINGLE WRITE
        9  TO END
```

```
***** ATTENTION FUNCTIONS *****
LO  - LOOP
PO  - DISPLAY OFF/ON FOR FAST SCOPE LOOP
M   - MASK OFF/ON ERROR DISPLAY
E   - END LOOP / PROG.
```

1 TO READ ALL ID'S: When this option is selected, all the card slots associated with each Block of DIS will be interrogated one Block at a time starting with Block 0. As each Block or Board of DIS is completely displayed, you are asked if you want to view the next sequential Block. Entering 'N' or 'N' will return you to the main option menu. Entering 'YES' or 'Y' will display all the macrofunction cards found in block 1, 2 etc.

```
BLOCK = 0  MACRO = 0  ID = 0001 BIC
BLOCK = 0  MACRO = 1  ID = 0020 RAM
BLOCK = 0  MACRO = 2  ID = 0014 VIDEO
BLOCK = 0  MACRO = 3  ID = 0023 RS232
BLOCK = 0  MACRO = 4  ID = 0002 DO
```

BLOCK COMPLETE. WANT NEXT ? _

This is a display of all the macrofunction cards found on block 0. If the block was full you would be prompted to hit enter on the keyboard in order to display the rest of the macrofunction cards.

2 TO DO A SINGLE READ: Select this option when you need to read data from a particular macrofunction card. You are prompted to enter the macrofunction address, command tag (CTC), and a value from 0 - F (0 = 1) for a short loop count. After the parameters have been keyed in, press the 'ENTER' or 'SEND' key and the requested data will be displayed.

```
ENTER => BLOCK  MACRO-ADDR  CTC  COUNT = 0090
          |||'-----> COUNT      (0-F)
          |||'-----> CTC         "
          |||'-----> MACRO-ADDR  "
          |||'-----> BLOCK       "
```

```
DBI = 0080 0000 0000 1000 0000
ENTER X TO END _
```

When you are prompted to enter 'X' to end, if you do so, this will detach the program, else just press enter and you will be returned to the main menu. If you return to the main menu, you may note that there is an asterisk '*' next to the operation you just performed. If you want to repeat the same operation again just press enter twice and the operation will be repeated with the same parameters entered previously.

IBM Internal Use Only

Note : After pressing enter once, you will be prompted for new parameters. You may enter the new set of parameters at this time, or just press enter again and the previously used parameters will be used.

```
ENTER => BLOCK  MACRO-ADDR  CTC  COUNT = 0093
```

```
DBI = 0080 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
DBI = 0080 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
DBI = 0080 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
ENTER X TO END _
```

3 TO DO A SINGLE WRITE: This option is selected when a write operation is required. When selected, macrofunction address, command tag (CTC), and short loop count 0-F (0 = 1) are required at the prompt. This operation will write data you enter to the selected macrofunction card. The loop count, if greater than 1 will create a specific number of prompts allowing new CTC and DATA to be entered for each one. If only one CTC/DATA combination is required, just pressing enter for each of the prompts will execute the operation with the parameters previously entered.

If a fast, continuous loop is required, use the 'Attention' loop command instead. The 'Attention' functions are explained later in this section.

```
ENTER => BLOCK  MACRO-ADDR  CTC  COUNT = 0420
ENTER DATA TO WRITE = 0000 1234
|-----> Data entered by user
|-----> Data from a previous write
```

The following example illustrates a write with a count of 2.

```
ENTER => BLOCK  MACRO-ADDR  CTC  COUNT = 0422
ENTER DATA TO WRITE = 1234 FFFF
ENTER CTC = 0002 0003
ENTER DATA TO WRITE = FFFF 0055
```

Notice the first word of each prompt shows the DATA or CTC from the previous operation. The second word is the new DATA or CTC to be used. It is entered by the user. Also notice that when an odd CTC is entered, The odd or least significant byte of the data word is used. This is because when odd CTCs are issued in EDX/J, only one byte of data is transferred to the DIS. When even CTCs are issued, the corresponding odd CTC is also issued, transferring the entire 16 bit data word. Each write or read to/from the DIS will be preceded by an LSA select (CTC 0) sequence to readdress the card before the selected command is issued.

9 TO END: When this option is selected, the program will be detached. The program however will still remain in memory and can be restarted by using the Supervisor '\$S' command. It may be removed from memory by using the Supervisor '\$C' command, cancelling the program.

***** ATTENTION FUNCTIONS *****

For specific information on the 'Attention' key, see the section entitled 'Attention Processor'.

LO - LOOP: This function is used with the 'SINGLE READ', and 'SINGLE WRITE' options. This function when invoked, will create a continuous loop to the selected macrofunction card with the parameters you enter for that card. To start the loop function, press the 'Attention' key and press enter. This is necessary to get the Attention cursor on the screen. Then enter 'LO'. This should be done after the option 2 or 3 is selected to start the loop. To stop the loop press the 'Attention' key and enter 'E'. After you have entered the 'LO' command you will be allowed to enter the remaining parameters normally used with the 'READ' or 'WRITE' options. The loop will not start until all of the parameters have been entered.

Here is a key entry example for looping on a read operation starting from the main option menu.

IBM Internal Use Only

(type) 2
(press) The Attention key
(press) Enter
(type) L0
(press) Enter

After the previous key sequence is done, follow through the same way normally used for a 'SINGLE READ'. Then the loop will start.

If a read operation is being looped on, the data being read will be displayed at each read time. If any DIS errors occur during the loop you may or may not see them because the screen is being updated at a fast rate. To view errors only, see the 'P0' attention command.

If a write operation is being looped on, the display is not used unless an error occurs provided that errors are not masked off.

P0 - DISPLAY OFF/ON FOR FAST SCOPE LOOP: This command works in conjunction with the 'L0' and 'M' commands. It will only work while you are in 'Continuous Loop' mode. To use this command, press the 'Attention' key and enter the command 'P0'. This will inhibit the display of data to the screen. To enable the display, press the 'Attention' key and enter 'P0' again. Here are some application examples for the 'P0' function :

- Used when needing a hard loop on one card. This function turns off the data display which in turn inhibits the processor from talking to the Ram and Video Display macrofunctions. Only DIS errors will be displayed when encountered. This function speeds the normal loop up many times faster than when the Display is being updated.

To loop hard on one card only, the 'M' attention function may be used to inhibit error messages from being written to the Display cards if errors are occurring.

- Used to view intermittent errors as they occur. Normally, errors will be displayed unless masked off using the 'M' function.

M - MASK OFF/ON ERROR DISPLAY: This function turns off/on the display of DIS errors to the screen. When errors are being displayed, the frequency of loop repetitions to the card being looped on is cut by the occurrence of errors. Also it may be undesirable to have other operations occurring on the DIS interface, especially when trying to isolate a problem.

To use this function, press the 'Attention' key and enter the 'M' function. To unmask or enable the error display, repeat the operation.

E - END LOOP / PROG.: This function turns off 'continuous loop' mode, fast scope loop mode, and enables the error display. It also leaves you at a place where you can return to the main menu or detach the program.

NOTE: When the IBM 3101 Terminal is used as the system console, the Attention loop functions for display on/off will only affect the speed of the loop because the Video Display cards are not configured into the DIS and are therefore not used. Thus all operations on the macrofunction interface will be to the macrofunction being tested.

IBM Internal Use Only

ROC/AUTO-POLLER UTILITY (ROCDG)

This program/utility provides the user with a method of exercising Remote Operator Console (ROC) hardware. All ROC functions are supported. This utility will require that an Auto Poller DIS macrofunction be installed in the system along with the Auto Poller support card.

When this utility is loaded, the following menu will be displayed.

```
*****
*                                     *
*          AUTO POLLER/ROC DIAGNOSTIC OPTION MENU          *
*                                     *
***** REL 2.0 - mm/dd/yy-hh.mm *****

1-POLL LIST GENERATION      7-TABLE LIST GENERATION
2-XMIT DATA                8-WRITE/READ LOOP
3-READ DATA                9-READ STATUS
4-POLL MODE                 10-LIGHT TEST
5-CHANGE ADDRESS            11-END
6-AUTO TEST
```

ENTER OPTION #:

Note: Before proceeding, the correct address (LSA) of an Auto Poller card must have been entered. Otherwise the main menu will be redisplayed along with a message instructing the user to select option 5 to change the Auto Poller card address. Also recall that the Auto Poller card must be held reset (POR) for at least 1 microsecond after all voltages are nominal for proper operation. Thus the system must include adequate power-on-reset circuitry.

1-POLL LIST GENERATION: This option must be defined if the ROC(s) are to be polled for work. When selected, the user will be prompted for:

1. The number of terminals to be entered on the poll list and,
2. The terminal address (00H-FEH) for each terminal on the list.

IBM Internal Use Only

2-XMIT DATA: This option is used to write user specified data to a ROC. When selected, another menu will be displayed following a prompt for the terminal address.

ENTER ADDRESS OF ROC TERMINAL 01-FE : 11

ENTER 1 - TO WRITE DISPLAY
2 - TO READ DISPLAY
3 - TO WRITE ALARM

ENTER OPTION #: 1

ENTER 1 - WRITE DATA
2 - WRITE DATA-ENTRY MODE LOWER PANEL SHIFT
3 - WRITE DATA-FUNC MODE
4 - WRITE DATA-ENTRY MODE NO LOW PANEL SHIFT

ENTER OPTION #: 1

SPECIFY THE STARTING ADDRESS ON THE DISPLAY

ENTER CHARACTER LINE # 0,1,2, OR 3: 0

ENTER CHARACTER POSITION 0-F : 0

ENTER MSG: MESSAGE FOR ROC

The previous is an example of the steps needed to write a message to the ROC display. The remaining options are used in a similar fashion.

1 - TO WRITE DISPLAY

This option, when selected, will invoke the write submenu. From this, the user will be allowed to select from the following options, which of the "write modes" will be used.

1 - WRITE DATA Writes the ROC display and does not invoke a mode change.

2 - WRITE DATA-ENTRY MODE LOWER PANEL SHIFT
Write the ROC display and invoke data entry mode with the lower panel shifted.

3 - WRITE DATA-FUNC MODE
Write the ROC display and invoke function mode.

4 - WRITE DATA-ENTRY MODE NO LOWER PANEL SHIFT
Write the ROC display and invoke data entry mode with no shift on the lower panel.

2 - TO READ DISPLAY Read the entire contents of the ROC display buffer and display it.

3 - TO WRITE ALARM Sound the alarm on the ROC. This produces a short alarm burst.

3-TO READ DATA: This option reads the contents of the receive buffer. Before the receive buffer is read, the status register is interrogated. The read will take place only if the ROC is not in XMIT or RCV mode. Also, "interrupt pending" must be active.

4-POLL MODE: When selected, the following options will be provided.

1-POLL TERMINALS All terminals in the POLL LIST will be interrogated for work.

2-SUSPEND POLLING All polling is suspended.

IBM Internal Use Only

5-CHANGE ADDRESS: This option, when selected, provides the user with the ability of changing the Auto Poller card address (LSA).

```
AUTO POLLER LSA = 00xx          <= xx is the current LSA
DO YOU WISH TO ENTER A NEW LSA? Y
ENTER NEW LSA : 0B              <= block 0 macro B entered
```

6-AUTO TEST: This option places all of the terminals in function mode. At this point, if a function key on any of the ROCs is pressed, that terminal will be prompted for data entry. The data entered will be echoed back to the ROC display. The "EN" attention function will end this function and return the program to the main option menu.

7-TABLE LIST GENERATION: This option is used to load the character translation tables into the Auto Poller card. These tables must be installed if the Auto Poller is to automatically convert EBCDIC to ASCII for transmit and ASCII to EBCDIC for receive data.

8-WRITE,READ LOOP: This option, when selected, will loop, sending test messages to all terminals. Pressing a function button on a ROC will suspend this loop and place the terminals in function mode. Once in function mode, the result is the same as if the user selected the "AUTO TEST" option. The "EN" attention function will return the program to the main option menu. The time between displaying different data patterns on the ROC display may be varied using the "DL" attention function. If used, the time delay is entered in milliseconds.

9-READ STATUS: Reads the Auto Poller status register.

10-LIGHT TEST: This option will activate each of the ROC's LED button indicators in various sequences to provide a visual check of those indicators. The "EN" attention function will return the program to the main option menu.

11-END: Ends this program/utility.

IBM Internal Use Only

RS232 CARD TEST (RS232)

This test provides the ability to test the operation of the DIS RS232 macrofunction cards. This is a wrap test and therefore requires the cards' xmit data and xmit return to be jumpered to receive data and receive return respectively. This is best done at the 'D' shell cable connector at the end of the communication cable being used. It is best to check out the cards' operation over the length of cable in order to properly check the line drivers and receivers. The RS232 interface is made up of a two card set, both of which must be installed.

- RS232 Control Card. This card contains the SERDES function provided by the Universal Synchronous Asynchronous Receiver Transmitter (USART) module, 1024 bytes of receive and xmit buffer space, mode of operation selection, and the control interface used by the DIS channel.
- RS232 Support Card. This card contains the line drivers and receivers for EIA and current loop operation, and BAUD rate generators.

When this test is loaded the following menu will be displayed:

```
RS232J TEST INITIALIZED      mm/dd/yy
****  ATTN FUNCTIONS  ****
ER - LOOP ON ERRORS
EN - END LOOP / PROG.

* BUFFER FULL INTERRUPT IS DISPLAYED IN FIRST STATUS WORD

RS232  BLOCK + MACRO ADDR = 0003
ENTER A NEW ADDR TO CHANGE  ELSE HIT ENTER _
```

This program tests the RS232 cards in the following ways:

- Checks the ability of the card to transmit and receive various message lengths from 3 to 1024 bytes, incrementing the byte count after a list of selected data patterns have been successfully sent and received.
- The data patterns used insure that parity, if selected, will be alternated to provide a more effective test, and that each bit can be turned off and on.
- A CHECKSUM is calculated for each transmission and is checked when the data is received. Parity if selected, is also checked by the hardware as well.
- This test monitors the BICs interrupts and displays which interrupt occurs in the first word of the status information. This test does not rely on hardware interrupts to run. If your system uses one of the BIC interrupts for this card, you must check the display for the occurrence, and assignment of the proper interrupt bit.

The following illustrates an example of the display as the test runs. This test once started, will only stop if it gets an error, or if you stop it using the 'EN' attention function.

IBM Internal Use Only

```

*****
INITIAL STATUS + GOOD AND BAD COUNT : 0000 0090 0072 0000
TX WORD COUNT HIGH + LOW = 0040 0008
DATA AND COUNT : 0055 0055 00FF
START TRANSMIT - LOOKING FOR BUFFER FULL
STATUS = 0000 0098
INTBIT + STATUS, RCV WORD COUNT HIGH + LOW = 0009 00F0 0040 0008
DATA RCV: 0055 0055 00FF
*****
INITIAL STATUS + GOOD AND BAD COUNT : 0000 0090 0073 0000
TX WORD COUNT HIGH + LOW = 0040 0008
DATA AND COUNT : 00AA 00AA 00FF
START TRANSMIT - LOOKING FOR BUFFER FULL
STATUS = 0000 0098
INTBIT + STATUS, RCV WORD COUNT HIGH + LOW = 0009 00F0 0040 0008
DATA RCV: 00AA 00AA 00FF

```

The diagram shows a timing sequence for the RS232 test display. It includes signals for Data Pattern, Data Word Count, BIC Interrupt Bit, Card Status Reg, Word Count High, and Word Count Low. The signals are represented by horizontal lines with vertical markers indicating specific points in time.

Figure 3. RS232 Test Display.

The display is continually updated as the test runs, displaying status, word count, and data pattern information. Error codes are displayed as the errors occur with a description of the error condition.

ER: The 'ER' attention function is used to force the program into a loop even when errors are occurring. This may be useful when needing to troubleshoot under the failing condition. For information on the use of attention functions see the section on the 'Attention Processor'.

```

XMIT      2<
RECEIVE   3<
XMIT. RET 7<
RCV RET   18<
(+) CTS    5<
(+) DSR(DTR) 6<
(+) 5V TIE-UP 24<

```

Figure 4. 25 pin 'D' shell connections for RS232 wrap connector.

IBM Internal Use Only

APPENDIX A. DISK UTILITIES

DISK UTILITY PROGRAM (DUT)

This is an EDX/J program which provides the user with functions for management of the 'Native' disk hardware on a DSC. The following is an example of the DUT main menu.

```
***** ASPEN7 **
*   D I S K   U T I L I T Y   C O M M A N D S   *
*****mm/dd/yy-hh.mm*****

AL  ALLOCATE          CD  COMPARE DATASETS
CM  COMPRESS          CP  COPY DATASET
CA  COPY ALL DATASETS CS  COPY S/1 DATASET
DE  DELETE            DU  DUMP
FM  FORMAT FILE       PA  PATCH
QU  QUIESCE           RF  READ FREE SPACE ENTRY
RD  READ              RE  RENAME
RH  READ HEADER       SM  SCROLL MEMORY
MV  MOVE DATASET      WR  WRITE
VO  VARY ON DISPLAY
RV OR DIR  READ VOLUME DIRECTORY
EN  END
```

ENTER COMMAND =>

Figure 5. Disk Utility Option Menu.

Upon loading of the disk utility, the user is presented with a menu of available commands. To execute a desired function, the user must enter the desired command on the command prompt line. The user will then be prompted to enter additional information that the selected function requires. As the user becomes more experienced, he may in most cases, preenter information required by a given function.

IBM Internal Use Only

Dataset Qualification

Datasets may reside on up to four disk drives depending on the configuration of the DSC being used. The disk drives, themselves, are identified by drive numbers, whereas disks or diskettes are identified by volume identifications. Datasets may be qualified at any time, by appending to the dataset name being specified, a comma followed by either a drive number or a volume identification. The following examples illustrate two ways of qualifying the dataset TEMP which resides on a diskette with volume id EDX001 inserted in disk drive #1.

- TEMP,EDX001
- TEMP,1

Note that this dataset qualification facility prohibits the use of a comma as part of a dataset name. Some disk utility functions do not require any dataset qualification. In these cases, drives will be searched in order for the first occurrence of the specified dataset. Other disk utility functions require qualification as to where a user specified dataset resides. In these cases, if the user does not qualify the dataset, he will be prompted to enter the drive number on which the specified dataset resides. A 0 drive number entered at this point acts as an escape, causing the option menu to be redisplayed. If an invalid drive number is specified, the user will be presented with a table indicating the volume identifications of the diskettes that are currently in each of the disk drives. If a disk drive is empty, the corresponding volume identification will be indicated by asterisks. This table is intended to assist the user in specifying the correct drive number. Note that the same table may be displayed at any time by issuing the 'V0' (vary on display) command.

IBM Internal Use Only

DUT AVAILABLE OPTIONS

AL ALLOCATE: Allocate (create) a new dataset on a specified drive/volume.

CA COPY ALL DATASETS: Copy all datasets from source to destination -- disk drives specified by the user.

CD COMPARE DATASET: Compare one dataset with another and display the differences between the two.

CM COMPRESS FILE: This option requires the availability of a second disk drive with enough space to hold the largest dataset on the volume to be compressed. Also there must not be a dataset named "DUT-TEMP" on the second disk. A scratch file by that name will be used during the compress.

CP COPY: Copy a dataset from a specified drive/volume to another drive/volume.

CS COPY S/1 DATASET: Copy a dataset from the Series/1 to a dataset on drive/volume at the DSC. This function uses the DSC 'Host' port to transfer the data from the Series/1 to the DSC.

DE DELETE: Delete a dataset from a specified drive/volume.

DU DUMP: Dump a specified number of words from a dataset to the display for viewing.

EN END: End the disk utility program.

FM FORMAT FILE: Format a specified drive/file. The drive number, disk type, and disk format, is required on selection of this option.

- Valid drive numbers are 1-4. (0 to exit).

- Valid disk types are:

1 Single sided diskette (diskette 1).

2 Double sided diskette (diskette 2).

2D Double sided double density diskette (diskette 2D).

HF Hard file (fixed disk).

- Disk formats of 128 or 256 byte sectors are valid.

Disk initialization follows the format operation. Here the user is prompted for the following information:

volume label Optional identification label. May be a maximum of 6 characters in length with a restriction that the first character must not be a numeric digit.

owner name Optional ownership label. May be a maximum of 12 characters in length.

directory size Defines the maximum number of dataset entries in the directory.

MV MOVE DATASET: Move a dataset from source to destination -- disk drives specified by the user.

PA PATCH: Patch location(s) within a specified dataset on disk/diskette.

QU QUIESCE: Release the display to other program sources. To regain control of the display, press the 'Attention Key', type 'DUT' <enter>.

RD READ: Read a specified number of words from a dataset into the DSC's memory and display the data.

IBM Internal Use Only

To read a physical location of a disk file: the user must use one of the following reserved dataset names:

- `$$$DRx`
- `$$VLx`

The x refers to the desired drive number (1 through 4). `$$$DRx` allows the user to access diskette files from cylinder 1 on, and hard files from cylinder 0, head 1 on. `$$VLx` allows the user to access an entire disk file. If the user enters one of these reserved dataset names, he will be prompted to enter a physical address. The entered physical address must be in the form CCHHRR, where:

CC refers to the cylinder,

HH refers to the head, and

RR refers to the record.

At least 5 digits comprising a physical address must be entered. For users familiar with disk file organization, this command can be used in conjunction with the WR (write to disk) command in order to patch disk files.

RE RENAME: Rename a dataset on a specified drive/volume.

RF READ FREE SPACE ENTRY: Read and display each free space entry from the specified drive/volume.

RH READ HEADER: Read the header of a specified dataset on disk/diskette and display it.

RV OR DIR READ VOLUME DIRECTORY: Read and display the Volume name and directory entries from a specified drive.

SM SCROLL MEMORY: Scroll through DSC memory starting from a given location.

VO VARY ON DISPLAY: Displays the attributes associated with each attached disk drive.

WR WRITE: Write a specified number of words from a given location in DSC memory to disk/diskette. The memory address from which data will be written can be user specified or defaulted to the address displayed by the last read (RD) command. The following is an example of how a disk could be patched using the RD and WR commands:

1. read, using the RD command, the area on disk to be patched (use a dataset name of either `$$VLx` or `$$$DRx`).
2. note the memory address at which the area on disk is stored.
3. hit the PF7 key to enter Debug.
4. using the memory display and patch commands, alter the data placed in memory by the RD command.
5. quit debug, thereby returning to the disk utility application.
6. issue the write (WR) command, and specify the same dataset name and physical address as was specified in the RD command.
7. accept the displayed default address by hitting enter when prompted to enter a start address.
8. the disk utility program will then write the patched area of memory to disk.

This command is intended for users familiar with disk file organization who wish to patch non-dataset areas of a disk file.

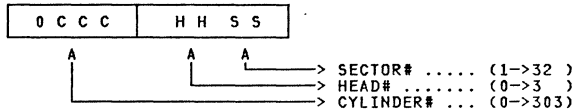
IBM Internal Use Only

FIELD DEFINITIONS FOR (RV OR DIR) DISPLAY

DCE - Directory Control Entry
DME - Directory Member Entry
FSE - Free Space Entry
#REC - Total number of records
#FRE - Total number of free records
#MEM - Current number of members
#FSE - Current number of free space entries
#DRE - Total number of directory records
#USR - Current number of free directory bytes
FREC - First logical record (256 bytes/record)
LREC - Last logical record (256 bytes/record)
TYPE - Dataset organization (Program = 3, Data = 1)
PLP - Program load point
BSIZ - Number of Bytes (TEXT) in dataset
RLDS - Number of words of relocation data
FCCCHHRR - Disk address designation (First Cylinder/Head/Record)
LCCCHHRR - Disk address designation (Last Cylinder/Head/Record)

RECORDS = SECTORS

CTS = CYLINDER / TRACK / SECTOR



VOLUME NAME ==> EDX001

DCE:	#REC	#FRE	#MEM	#FSE	#DRE	#USR
	000003B5	0000024B	0007	0001	000D	0BF8

DME NAMES:	FREC	LREC	TYPE	PLP	BSIZ	RLDS	FCCCHHRR	LCCCHHRR
\$EDXIPL	0000000E	000000C2	0003	0000	AD00	03D2	00020001	000F000C
DUTZZ	000000C3	000000F1	0003	0000	2506	0445	000F000D	00130007
MFTJ	000000F2	00000100	0003	0000	0B6C	0102	00130008	00140009
LOAD5	00000101	0000011C	0003	0000	101C	055A	0014000A	0016000B
LD5	0000011D	00000137	0003	0000	0FF0	0559	0016000C	0018000C
APUT	00000138	00000141	0003	0000	08B0	0071	0018000D	00190009
APDBX	00000142	0000016A	0003	0000	222E	02C5	0019000A	001C000B

ENTER COMMAND ==>

Figure 6. 'RV' (READ VOLUME) function display example.

IBM Internal Use Only

The following diagram displays the format profile for each of the diskette types and the hard file.

D I S K F O R M A T S				
	TYPE 1	TYPE 2	TYPE 2D	HARD FILE
BYTES/SECTOR	128 256	128 256	128 256	256
SECTORS/HEAD	26 15	26 15	52 30	32
LRECS/HEAD	13 15	13 15	26 30	32
BYTES/HEAD	1328 3840	3328 3840	6656 7680	8192
HEADS/CYL	1	2	2	4
LRECS/CYL	13 15	26 30	52 60	128
BYTES/CYL	3328 3840	6656 7680	13,312 / 15,360	32768
CYLS/VOLUME	74	74	74	303
LRECS/VOLUME	962 1110	1924 2220	3848 4440	38784
BYTES/VOLUME	246,272 / 284,160	492,544 / 568,320	985,088 / 1,136,640	9,928,704

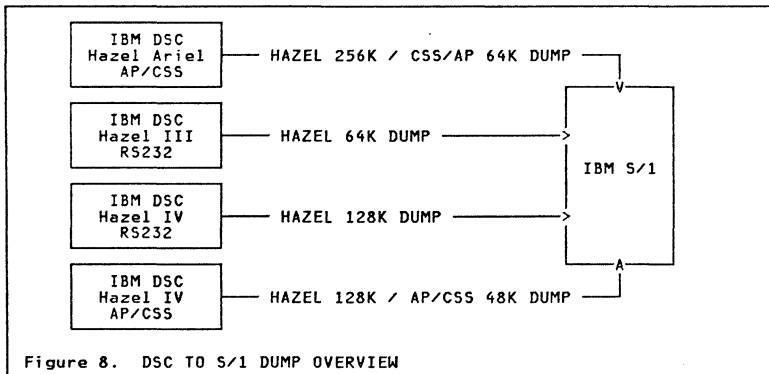
Figure 7. Disk/Diskette Formats and Types.

IBM Internal Use Only

APPENDIX B. DSC MEMORY DUMP FACILITY

MEMORY DUMP TO S/1 DISKETTE

This document explains how to dump the DSC memory to S/1. First it describes how to initialize diskettes for the DSC to S/1 dump. Then it shows how to start a dump from a DSC with RS232 and AP/CSS based communications. Finally, it describes how to use DTD (Dump to Diskette) and UTDH (Utility to Display or Dump to Host) S/1 utility programs to dump DSC memory to host and display the dumps on S/1 console or a local printer.



IBM Internal Use Only

DISKETTE INITIALIZATION

The S/1 utilities used for diskette initialization are: \$DASDI, \$INITDSK and \$DISKUT1. The description of the initialization consists of prompts from the S/1 utility programs and the responses entered by the user. The user responses are highlighted.

Note, that Hazel/Ariel memory dump (256K) requires 2D type diskettes.

\$DASDI and \$INITDSK

- > \$L \$DASDI
- ENTER DISK INITIALIZATION OPTION: 1
- ENTER DISKETTE ADDRESS IN HEX: 2 or ?²
- INITIALIZE FOR USE WITH THE IBM EVENT DRIVEN EXECUTIVE (Y/N)? Y
- SELECT SECTOR SIZE (1=128, 2=256): 2
- FORMATTING WILL DESTROY ALL DATA ON THE DISKETTE.
- CONTINUE (Y/N)? Y or N³
- LOAD \$INITDSK (Y/N)? Y
- COMMAND (?) ID
- DEVICE ADDRESS: 2
- INITIALIZE DEVICE MAY DESTROY ALL DATA CONTINUE (Y/N)? Y or N³
- INITIALIZE DISKETTE AS MULTI-VOLUME TYPE (Y/N)? N
- NEW VOLUME LABEL: EDXDMP
- ENTER OWNER IDENTIFICATION: owner name and/or initials
- SINGLE-VOLUME TYPE DISKETTE INITIALIZED
- INITIALIZE THE VOLUME (Y/N)? Y
- MAXIMUM NUMBER OF DATA SETS: 102
- DO YOU WANT WRITE VERIFY FOR THIS VOLUME (Y/N)? Y
- ALLOCATE \$EDXNUC (Y/N)? N
- COMMAND (?) EN
- ANOTHER DISKETTE? N
- ENTER DISK INITIALIZATION OPTION: 4

\$DISKUT1

² The diskette address may not be 2 for your installation.

³ To avoid erasure of data from the hard disk listen for the diskette drive to recalibrate. If you do not hear the diskette drive, enter N to abort the initialization!

IBM Internal Use Only

- > \$L \$DISKUT1
- COMMAND (?) CV EDXDMP
- COMMAND (?) AL
- MEMBER NAME: DMPDIR
- HOW MANY RECORDS? 2
- DEFAULT TYPE = DATA - OK? Y
- COMMAND (?) AL
- MEMBER NAME: DUMP
- HOW MANY RECORDS? 1028 or 2056⁴
- DEFAULT TYPE = DATA - OK? Y
- COMMAND (?) EN

DISABLING S/1 SENDS TO DSC

A requirement for a successful DSC to S/1 dump is that there be no other communication traffic between the DSC and the S/1 (except for the dump). Therefore, before the dump is started, the S/1 DTD program "disables" the line via which the dump is to take place. When a line is "disabled" the TP Manager in S/1 prevents S/1 application programs to send to the "disabled" line, messages with priority higher than 10. The priority of messages sent from the DTD program is 9.

DSC MEMORY DUMP AND DISPLAY

STARTING DSC TO S/1 DUMP

PROCEDURAL RULES

1. Only one dump per EDXDMP diskette is allowed.
2. Reuse of EDXDMP diskettes does not require reformatting.
3. To aid in the maintenance of EDXDMP diskettes the following measures are recommended:
 - Maintain a large supply (say 10) of clearly labeled EDXDMP diskettes in a box posted on a wall.
 - All the used EDXDMP diskettes should have up to date labels and be kept for future reference in a separate box.
4. It is strongly advised that a copy of this document be posted on the wall in the vicinity of the DSC's and the S/1's.

⁴ Hazel/Ariel requires 2056 records Hazel/3 and Hazel/4 require only 1028 records. Note, that 2D type diskette is required to allocate 2056 records.

IBM Internal Use Only

DUMP PROCEDURE FOR RS232 BASED COMMUNICATIONS

To initiate the DSC to S/1 dump, perform the following steps in sequence:

1. Determine the line number of the DSC which is required by the S/1 DTD program.
2. Insert EDXDMP diskette into S/1 diskette drive.
3. Load the S/1 DTD program and enter requested information.
4. When the DTD program prompts to start the dump on DSC go through the following steps:
 - If not already in the M/C handler, hit TEST REQ key on DSC keyboard to enter the M/C handler.
 - Hit PF8 key to arm DSC for the dump.
 - Observe the light pattern on front panel of the DSC - lights 0 to 3 should be off and lights 4 to 7 should start flashing. Flushing lights indicate that Hazel is waiting for the S/1 DTD program to initiate the dump.
 - Any other light pattern on the front panel indicates a RS232 initialization error; hit the following keys: PF9 and PF8 to re-start the dump.
5. Continue entering information requested by the S/1 DTD program.

TERMINATION / RESTART FOR RS232 BASED DUMP

When the dump ends, the dump program in DSC returns control to the M/C handler.

Hit PF9 in case the dump abends, or to terminate the dump in progress. The PF9 key disarms the dump and returns control to the M/C handler.

IBM Internal Use Only

DUMP PROCEDURE FOR AP/CSS BASED COMMUNICATIONS

To initiate the DSC to S/I dump, perform the following steps in sequence:

1. Determine the line number of the DSC which is required by the S/I DTD program.
2. Insert EDXDMP diskette into S/I diskette drive.
3. Load the S/I DTD program and enter requested information.
4. When the DTD program prompts to start the dump on DSC go through the following steps:
 - Hit PF1 key on the 3101 keyboard to enter AP Hazel Debug.
 - Enter HA to dump Hazel memory or AP to dump AP/CSS memory.
 - Enter DU to arm the dump. At this point AP is waiting for the S/I DTD program to initiate the dump.
 - To determine that the dump is armed, observe the AP light pattern - lights 0 to 3 should start flashing. Flushing lights indicate that the AP is waiting for the S/I DTD program to initiate the dump.
5. Continue entering information requested by the S/I DTD program.

TERMINATION / RESTART FOR AP/CSS BASED DUMP

When the dump ends, the dump program in AP returns control to the AP Hazel Debug.

Hit ENTER in case the dump abends, or to terminate the dump in progress. The ENTER ENTER key disarms the dump and returns control to Hazel Debug.

IBM Internal Use Only

DISPLAYING/UPLOADING THE DUMP

Using the S/1 UTDH (Utility to Display or Dump to Host) program, the DSC dump may be displayed on S/1 console, printed on S/1 printer or uploaded to S/370.

To use the UTDH program the following steps are required:

1. Insert EDXDMP diskette into S/1 diskette drive.
2. Load UTDH program.
3. Enter '?' to display available commands.

ALLOCATING DUMP DATA SET ON S/370

If the DSC dump is to be sent to the host, a dump data set must be allocated on S/370. The following is an example of how to allocate a sequential dump data set using ISPF 3.2:

```
SPACE UNITS      ==> BLKS
PRIMARY QUANT    ==> 105
SECONDARY QUANT  ==> 2
DIRECTORY BLOCKS ==> 0
RECORD FORMAT    ==> FB
RECORD LENGTH    ==> 256
BLOCK SIZE       ==> 2560
```


IBM Internal Use Only

FORMATTING THE DUMP ON S/370

To format the dump data set uploaded from S/1 use the DSCDUMP clist. The clist accepts six optional key words: four with parameters and two without parameters.

The following is a list of keywords with parameters:

1. DA(name) - where name is the second qualifier of the dump data set. DEFAULT = DSCDUMP.
2. SYSOUT(sysout) - where sysout is the sysout class which defines the type of paper on which the formatted dump will be printed. DEFAULT = A.
3. LOADLIB(loadlib) - where loadlib is the fully qualified name of the data set containing the dump formatting load module. DEFAULT = ts85834.PLI.LOAD.
4. LOADMEM(loadmem) - where loadmem is the name of the the dump formatting load module. DEFAULT = FRMTDUMP.

The following is a list of keywords without parameters:

1. HOLD - when specified causes the formatted dump to be held on the output queue. The user may then browse it, purge it, or requeue it to the printer.
2. DIAG - when specified activates MSG, LIST, CONLIST and SYMLIST clist processing options.

IBM MAINTENANCE DEVICE ATTACHMENT TO DSS (STANDALONE SYSTEMS)

This section provides a basic description of the IBM Maintenance Device and operating procedures for normal use.

The IBM Maintenance Device (MD) is basically a microcomputer with an 8 inch Floppy Disk drive and is configured with several communication ports. The entire unit is contained in a suitcase. The applications adapted to this unit for DSS equipment are as follows.

1. Load a nucleus from the MD into the DSC.
2. Load an EDX/J application program from the MD into the DSC.
3. Alter the DSC memory via MD.
4. Copy the DSC memory onto Diskette via MD (DUMP).

IBM Internal Use Only

DUMP PROCEDURE (MD)

1. Load the MD DUMP diskette into the drive.
2. Turn the POWER ON to the MD (you must have properly loaded the DUMP diskette at this time).
3. Connect the Programmable I/O Maintenance Device Adapter (PIOMDA) cable to the associated female connector on the DSC.

Watch the KEYBOARD/DISPLAY. You will notice that the MD is running some internal diagnostics on itself. If an error occurs on any of the diagnostic tests, try another DUMP diskette (also verifying that the diskette is loaded properly in the drive). If the problem still exists get another MD to use until the other one can be repaired.

When the MD internal diagnostic tests have finished, you will be prompted to 'OPEN' the communication port between the DSC and the MD.

4. Press 'ENTER' on the KEYBOARD/DISPLAY to 'OPEN' the PIOMDA port.

At this time the KEYBOARD/DISPLAY should read 'PIOMDA PORT OPENED HIT ENTER ON THE DSC KEYBOARD'.

5. Press 'ENTER' on the DSC KEYBOARD, then press 'ENTER' on the MD KEYBOARD.

The MD KEYBOARD/DISPLAY should read 'MDIC READY FOR USE' if you have established communication with the DSC. If not, check all MD cable connections to the DSC. NOTE: When you pressed 'ENTER' on the DSC keyboard, the DSC's display should not have changed, and the flashing PROCESS CHK indicator should have stopped flashing. If this is true, then you have established communication with the DSC.

6. Press 'ENTER' on the MD KEYBOARD/DISPLAY to continue.
7. Select option '3...DUMP DSC MEMORY' from the option menu and press 'ENTER'.
8. Press 'ENTER' on the MD KEYBOARD/DISPLAY again to enter '0000' for the DSC DUMP memory start address.

At this time the MD should be storing the data from the DSC's memory onto the DUMP diskette. This process will take approximately 10 minutes.

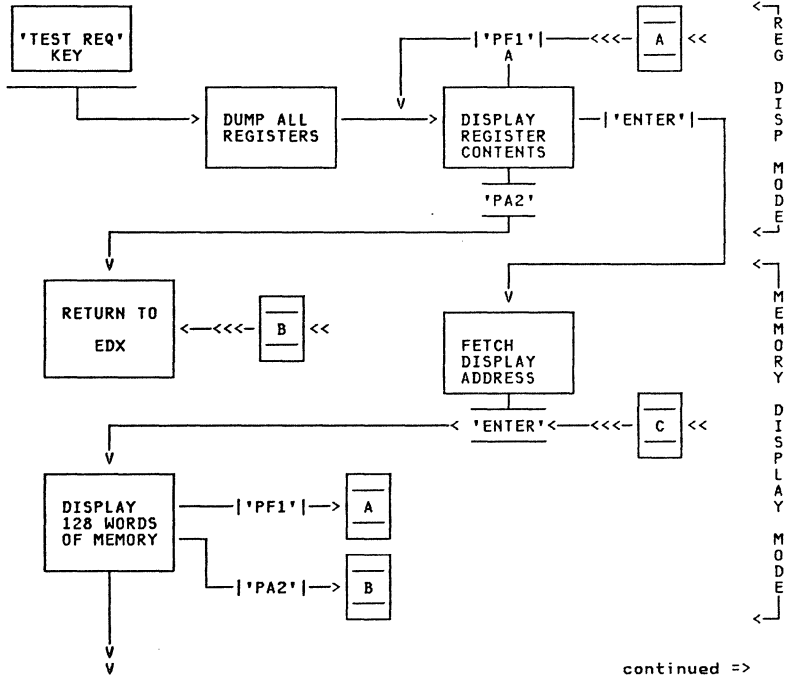
9. Press 'ENTER' on the MD KEYBOARD/DISPLAY to return to the main option menu.
10. Select option '5...END' from the option menu and press 'ENTER' to disconnect the MD from the DSC.
11. Remove the MD DUMP diskette and forward it to your support group for analysis.

The DSC may be IPLed with your system diagnostics now to find any detectable hardware problems. It is important to DUMP the memory to diskette first so a machine history may be built in case the problem turns out to be intermittent.

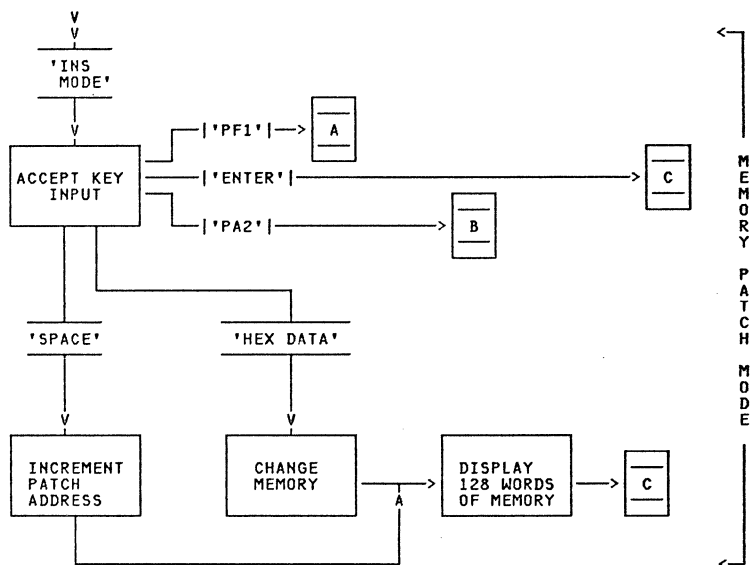
APPENDIX C. DEBUG UTILITIES

TEST REQ KEY SUPPORT

This section describes the debug functions supported when using the 3277 keyboard with standard video monitor for the system console. These functions are accessed by pressing the 'TEST REQ' key on the console.



IBM Internal Use Only



TO THE USER: The 'TEST KEY' support masks off all interrupts to the system and suspends the execution of all EDX/J programs and support. Thus it may be undesirable to invoke the 'TEST KEY' functions when EDX/J programs must be synchronized with the external environment i.e. 'REAL-TIME'. The 'PF1' key is used to invoke these same functions during a machine check condition (FLASHING PROCESS CHECK INDICATOR).

IBM Internal Use Only

REGISTER DISPLAY MODE

LOCAL STORAGE REGISTER (LSR): The LSR is a general purpose register. All LSRs are displayed in the first 8 lines of the REGISTER DISPLAY (DSC II,III). DSC IV systems are configured with 4 times the number of LSRs. The primary register display will show some of these, but all may be displayed using EXTENDED REGISTER DISPLAY mode.

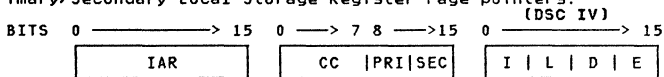
DSC II,III Interrupt Level Assignments:

<u>Interrupt Level</u>	<u>Use</u>
0	Machine Check Support
1	TOD/Interval Timer Support
2	Native Keyboard Support
3	DIS Support
4	DIS/Host Support
5	EDX/J Interpreter
6	N/A
7	N/A

DSC IV Interrupt Level Assignments:

<u>Interrupt Level</u>	<u>Use</u>
0	Machine Check Support
1	TOD/Interval Timer Support
2	Native Keyboard Support
3	DIS Support
4	Attached Processors and Host Support
5	N/A
6	EDX/J Interpreter
7	N/A

PROGRAM STATUS WORD (PSW): The EDX/J operating system uses 6 of the 8 hardware/program levels. Each of the 8 levels has its own 2-word PSW (DSC II,III) or 3-word PSW (DSC IV) used to save the level's 'NEXT SEQUENTIAL INSTRUCTION POINTER' (IAR), its Condition Codes, and its Primary/Secondary Local Storage Register Page pointers.



The I, L, D, and E external registers are used for controlling memory and register partitions. Only the rightmost nibble (4-bits) of these registers are displayed. See Hazel IV hardware documentation.

IBM Internal Use Only

JIB PRIME CONDITION CODES (CC) :

<u>Bit</u>	<u>Description</u>
0	Memory Parity Error.....(MACHINE CHECK)
1	Register Parity Error (LSR/EXT).....(MACHINE CHECK)
2	Internal Error (BR DECISION/CLK etc.)..(MACHINE CHECK)
3	PSW Swap Latch
4	Micro Instruction Register Bit-0
5	Equal/All
6	Zero/None
7	Carry

EXTERNAL REGISTERS (XRs): The XRs form the 9-bit (8 data + 1 parity) medium for transferring information to/from the machine. These registers are described in the HAZEL functional specification for each of the HAZEL processors (II, III, OR IV).

REGISTERS OF SPECIAL INTEREST: Certain registers in the machine contain pointers into the currently running EDX/J program. These registers may be viewed by the user and are of great value to all debug operations. The Local Storage Registers referred to are displayed on the "EDX/J INTERPRETER" interrupt level. See "Interrupt Level assignments" at the beginning of this section and the following register displays for the particular locations of these registers. Here is a list of these registers and their descriptions.

<u>Registers</u>	<u>Description</u>
------------------	--------------------

A,8	Local Storage Registers used as the current EDX/J instruction pointer (IAR).
6,7	Local Storage Registers used as the pointer to the current EDX/J Task Control Block (TCB).

18 (decimal 24) (DSC IV ONLY)

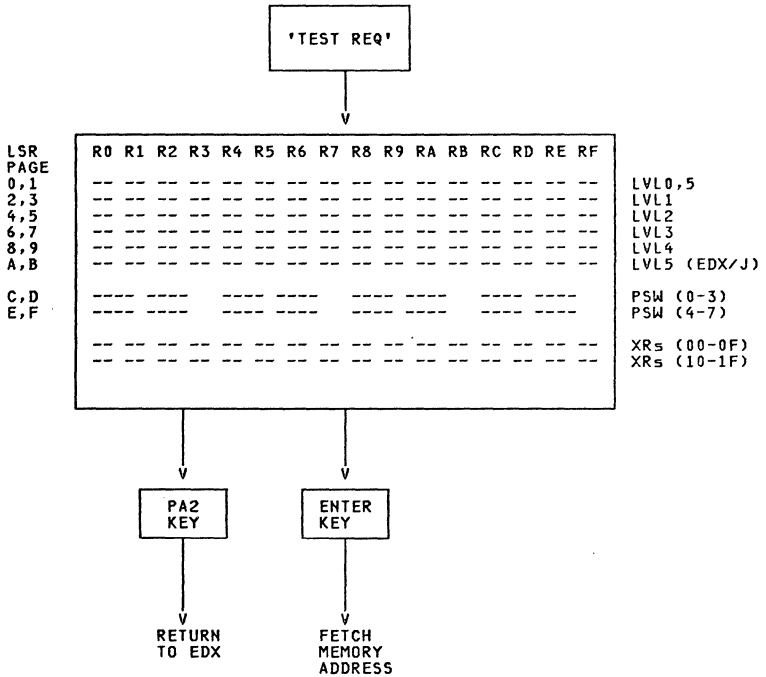
"I" Register. External Register used for selecting the current "High Memory" partition (address 8000H to FFFFH). Valid settings are:

0	Select memory partitions 0 & 1.
1	Select memory partitions 0 & 2.
2	Select memory partitions 0 & 3.
3-6	Valid for machines with the extended memory feature-- selects partitions 4 through 7.

Note: Partition "0" (address 0000H to 7FFFH) is "Fixed".

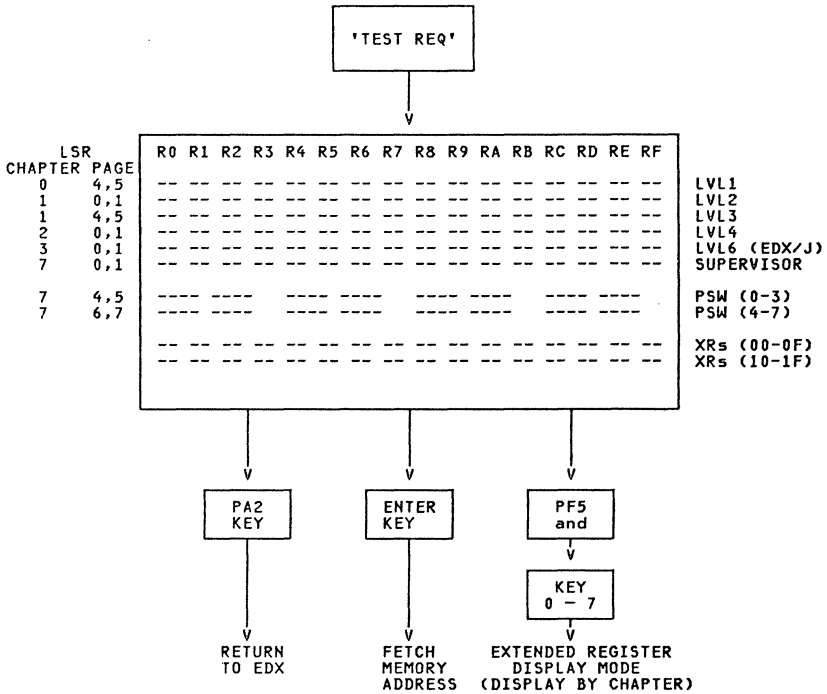
IBM Internal Use Only

----- REGISTER DISPLAY MODE -----
(DSC II,III)



IBM Internal Use Only

----- REGISTER DISPLAY MODE -----
(DSC IV)



----- EXTENDED REGISTER DISPLAY MODE -----
(DISPLAY REGISTERS BY CHAPTER/BLOCK)
(DSC IV ONLY)



IBM Internal Use Only

MEMORY DISPLAY MODE

Memory display mode has two phases, (A) fetching the display start address and (B) displaying memory from that address.

FETCH DISPLAY ADDRESS PHASE: On entering 'Memory Display Mode' the user is prompted to enter a memory address. Once the user has keyed in a memory address the 'ENTER' key is pressed to initiate the displaying of the contents of memory starting at the address given.

NOTE: If the user doesn't enter a (new) address, and just presses the 'ENTER' key, the previously entered address (or zero) is used as the display's start address.

VALID KEYS

HEXADECIMAL keys (0-F)

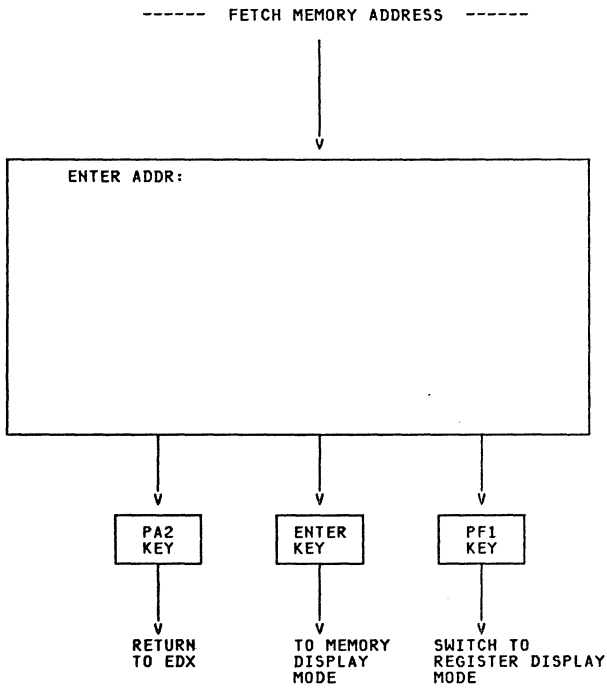
All HEX character keys are valid as entry to the address prompt. Only the last 4 HEX key entries are used in the development of the memory address.

ENTER The 'ENTER' key is used to indicate that a memory start address has been entered or default address accepted. A full screen display of 128 words of memory starting at the memory address requested is initiated.

PF1 The 'PF1' key switches the display back to 'REGISTER MODE'.

PA2 The 'PA2' key exits from the 'TEST REQ' function mode and returns to EDX.

IBM Internal Use Only



IBM Internal Use Only

DISPLAY FULL SCREEN PHASE: The full screen displays 128 words of memory starting at the address previously entered. The data layout on the screen consists of 9 columns and 16 rows.

The leftmost column contains the memory address for the start of each row of data. The other 8 words of data in the row display the contents of memory starting with the memory address shown in column 1 of the row.

VALID KEYS

SCROLL UP The 'SCROLL UP' key will increment the current screen's start address by 128 and initiate a full screen display of the next 128 words of higher memory.

SCROLL DOWN
The 'SCROLL DOWN' key will decrement the current screen's start address by 128 and initiate a full screen display of the preceeding 128 words of memory.

SCROLL LEFT
The 'SCROLL LEFT' key will decrement the current screen's start address by 1 and perform a new full screen display.

SCROLL RIGHT
The 'SCROLL RIGHT' key will increment the current screen's start address by 1 and perform a new full screen display.

LINE ADVANCE
The 'LINE ADVANCE' key will increment the current screen's start address by 8 and perform a new full screen display.

PF6 (DSC IV ONLY)
The PF6 key followed by decimal input 1-3 is used to display the memory partition 1, 2, or 3 respectively. These partitions all start at the 8000 (HEX) address boundary. Partition "0" is located from address 0000 to 7FFF (HEX).

PF7
The 'PF7' key stores FFFF (HEX) at the memory start address. This data pattern will act as an EDX/J break point and can be used for program debug. When EDX detects this pattern, execution stops and a "soft" machine check condition is entered. Pressing 'PF7' once more at the same start address restores the original contents of that location.

PF10
The 'PF10' key like PF7, stores a "hang" code at the memory start address. This, however, is a JIB hang, and is only used when debugging JIB assembler code routines. Pressing the 'PF10' key once more at the same memory start address, restores the original contents of that location.

IBM Internal Use Only

HEXADECIMAL keys (0-F)

All 'HEX' character keys are valid. If the user wishes to display a new memory address (without scrolling), just enter the address. The screen support will return to the 'FETCH ADDRESS MODE' of operation. When 'ENTER' is pressed, 128 words of memory will be displayed starting from the address entered.

ESCAPE DISPLAY MODE KEYS

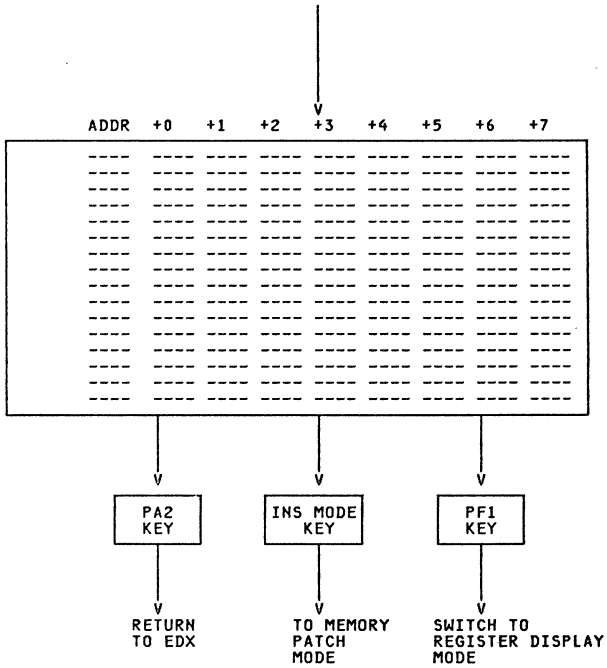
PF1 The 'PF1' key switches the display to 'REGISTER DISPLAY MODE'.

PA2 The 'PA2' key exits from the 'TEST REQ' function mode and returns to EDX.

INS MODE The 'INS MODE' key is used to enter 'PATCH MEMORY' mode.

IBM Internal Use Only

----- MEMORY DISPLAY MODE -----



IBM Internal Use Only

MEMORY PATCH MODE: Memory patch mode is indicated when the upper left hand corner of the screen shows a patch address in addition to the full screen display. The initial patch address is assumed to be the same as the current start address used in display memory mode.

VALID PATCH MEMORY KEYS

HEXADECIMAL keys (0-F)

All 'HEX' character keys are valid. These keys are now used to change the contents of the memory location addressed by the 'P=XXXX' prompt message pointer. Each key inputs a 'HEX' value which is shifted into the patch address memory location. A full screen display is initiated to show the current state of the patched memory location. Only the last four 'HEX' character entries are used in patching a word of memory.

SPACE The 'SPACE' bar will increment the patch address by 1 and initiates a full screen display. The contents of the word 'Advanced From' are either left intact (if not patched by 'HEX' key entries), or left with the final patch data entered. The space bar can therefore be used to skip over locations which don't require change and advance the patch address pointer to the next memory address requiring change.

TAB The 'TAB' key will increment the patch pointer address by 1 and store the current patch data word into that address. The 'TAB' key auto-repeats, providing a convenient method of clearing or filling parts of memory with a constant.

ESCAPE PATCH MODE KEYS

SCROLL keys (all)

Any of the 'SCROLL' keys will cause an exit from patch mode back to memory display mode.

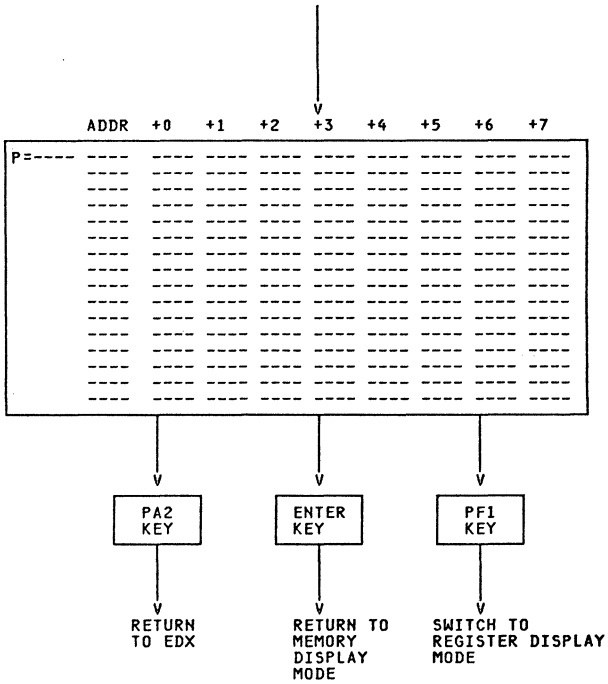
ENTER The 'ENTER' key will store patch data and exit to memory display mode.

PF1 The 'PF1' key switches the display to 'REGISTER DISPLAY MODE'.

PA2 The 'PA2' key exits from the 'TEST REQ' function mode and returns to EDX.

IBM Internal Use Only

```
----- MEMORY PATCH MODE -----
```



IBM Internal Use Only

3101 TERMINAL DEBUG SUPPORT

HAZEL Debug Tool Overview

The Hazel Debug Tool (Debug) is available as part of the console AP's operating system. This tool provides the user with a comprehensive set of functions which will assist in development and debugging of both EDX and JIB code. Once the DSC has been IPLed, the user may access Debug at any time by hitting the PF1 key. If the active task is sending information to the screen, the output operation will be completed before Debug is invoked. If, however, the active task is waiting for user input to the screen, the input operation will be enqueued, and Debug will be invoked immediately. Once Debug has been activated, it will assume control of the screen. An active task involving screen I/O will have its execution suspended until Debug has released control of the screen.

If two screens are attached to the DSC being used, the application being tested may use one screen, and Debug the other. In this case, an application program involving screen I/O will not have to compete with Debug for the use of a screen. Debug will be activated on the terminal on which the PF1 key was hit.

Upon initial entry to Debug, the user is presented with a menu of primary commands (refer to Primary Commands Menu). Debug is driven by three types of commands: primary commands, secondary commands, and global commands. Primary commands initiate different types of activities. Examples of available activities include display and update of memory, display and update of registers, and stopping and tracing EDX or JIB code. Secondary commands are "second level" commands which perform functions that are specific to the activity selected by a primary command. These commands can not be accessed directly, but must be preceded by the appropriate primary command. Global commands set global parameters within Debug. These parameters define a format for screen output, as well as an area of HAZEL or AP memory to which memory oriented commands apply.

Most users will only be interested in a subset of the available functions. For users interested in operating system development involving JIB assembler code, Debug provides commands that will stop, single step assembler code, commands that will display and alter the contents of local storage and external registers, and a command that will display the contents of the system queues. For a description of the hardware relevant to development of this nature, refer to "DSC IV HARDWARE SPECIFICATIONS". Users only interested in development of EDX code can ignore these functions entirely.

To become adept at using this tool, the user is encouraged to actually try the commands as they are covered in this manual.

IBM Internal Use Only

Command Entry

All functions are initiated by user entered commands with or without associated parameters. Commands are one or two characters in length, and parameters are either decimal or hexadecimal numbers up to four digits in length. Whether a parameter is to be entered as a decimal or hexadecimal number depends upon the accompanying command. Debug automatically pads zeros to the left of hexadecimal parameters less than four digits in length. Both commands and parameters may be entered with any number of intervening blanks as long as no ambiguity results. The following examples illustrate command and parameter entry and interpretation:

1.

```
PA 0000 0000    patches memory location x'0000' with the pattern:
                  0000.
P A 0000 0000    patches 3 implied sequential memory locations with the
                  pattern: 000A 0000 0000 (refer to the "patch memory
                  at current location" memory command).
PABCD 0000       patches memory location x'0BCD' with the pattern:
                  0000.
```

2.

```
M 9000          displays HAZEL memory at memory location x'9000'.
M9000           displays HAZEL memory at memory location x'9000'.
```

Special Prompt Characters

At any time, the user may use addition "+" and subtraction "-" sign between parameters to form arithmetic expressions. When Debug interprets the commands and parameters entered by the user, it generates a single equivalent parameter for each arithmetic expression it encounters. This capability is useful when displaying or patching memory where offsets are involved. The following examples illustrate uses of special prompt characters:

OPERATION	EQUIVALENT
PA 8000+AA 0000	=> PA 80AA 0000
P 1000+5 605-5	=> P 1005 600
I -2	=> I FFFE

Consider transferring 4AA words of memory (see Transfer memory command) from an AP starting address of 8FA0, to an AP destination address of 6000. Since the Memory Transfer command requires source start and end addresses, memory transfer could be performed by entering:

<CPU>	<cmd>	<st_addr>	<end_addr>	<target>
AP	T	8FA0	8FA0+4AA-1	6000

IBM Internal Use Only

Global Commands

There are two types of global commands. One type is used to select an area of AP or HAZEL memory to which memory oriented commands apply. The other type is used to define a format for screen output. Global commands may be entered at any time; alone, or in conjunction with primary, secondary, or other global commands. The following is a list of global commands which define an area of memory:

CMD PARAMETER DESCRIPTION

HA	Switch to HAZEL mode. Focus all operations on the HAZEL CPU.
AP	Switch to AP mode. Focus all operations on the AP CPU.
CP <PARTN>	Change Partition. Address locations 0-7FFF are common to all three partitions. Valid partitions values range from 1 to 3.
B= <ADDRESS>	Set HAZEL memory base offset. The specified base address is added to any HAZEL address specified whether displaying HAZEL memory, or stopping EDX or JIB. Setting a base address is useful when debugging relocatable EDX programs.
DM <ADDRESS>	Display HAZEL memory on EDX stop. Unless the DM address is set to 0, on each EDX stop, stop information will include 8 words of HAZEL memory located at the specified address. Note that the base address specified by the "B=" command is added to the address specified the "DM" command.
TS	Switch to SMALL EDX trace buffer size (256 words in the nucleus).
TL	Switch to LARGE EDX trace buffer size (24,576 words in the user memory A000 - FFFF partition 2). WARNING: LARGE trace will overwrite user memory A000 - FFFF in partition 2.

The following is a list of global commands which define a format for screen output:

CMD PARAMETER DESCRIPTION

PM	Picture Mode. Display new screen output beginning at the top of the screen.
SM	Scroll Mode. Display new screen output beneath the output which was last displayed.
R.	Do not display unchanged registers on a JIB stop. Periods "." are displayed in place of register pairs that have not changed since the previous JIB stop.
R?	Display all registers on JIB stop.
SK <#LINES>	Skip line. Redisplay the command prompt line a specified number of lines up or down. Valid values range from -7 to 7.
SP <#LINES>	Set memory display spacing. This command sets the number of blank lines to be inserted between blocks of memory displayed under scroll mode. Valid values range from 0 to 7. Entering "SP" alone, defaults to a spacing of 1.
L <LINE_CNT>	Set decimal number of lines of memory to be displayed. Valid line counts range from 1 to 99. The default of 4 lines can be set by entering a single "L". This command also sets the word count (refer to the "W" command) to a corresponding number of words.

IBM Internal Use Only

W <WORD_CNT> Set decimal number of words of memory to be displayed. When this command is used, subsequent scrolling will be by the specified number of words. For example, a WORD_CNT of 1 will cause the block of memory displayed to shift by one memory location each time the ENTER key is pressed. A WORD_CNT of 0 is useful for continually redisplaying the same block of memory. The word count is reset by entering a "L" command, or a "W" command without any associated parameters.

KS/FS Keep/Free Debug screen on EDX/JIB stop. The "KS" command keeps the Debug screen, and prevents the application screen from being restored between EDX/JIB stops. This command should only be used when the code between EDX/JIB stops does not require screen I/O. When the "FS" command is issued, the application screen is restored between EDX/JIB stops.

NOTE: The current global settings can be displayed by issuing the "G=" primary command.

Primary Commands

Primary Commands Menu

```

- PRIMARY COMMANDS ----- HAZEL DEBUG -
M (ADDR)      - MEMORY DISPLAY          SQ - SYSTEM QUEUE DISPLAY
(ADDR)        - MEMORY DISPLAY          G= - GLOBAL COMMANDS MENU
R (XR_STACK)  - REGISTER DISPLAY        ? - AVAILABLE COMMANDS
DU            - DUMP TO S/I              Q - QUIT DEBUG
ET           - EDX TRACE DISPLAY         RD - REDISPLAY STOP INFO
IP           - HAZEL SOFT IPL            T - STOP TITLES DISPLAY

* SJ (ADDR)(CNT)(ITER)                  - STOP JIB
* SE (ADDR)(TCB)(CNT)(ITER)              - STOP EDX
* TE (ST_ADDR)(TCB)(CNT)(ITER)(EN_ADDR)  - TRACE EDX

*   USE "0" TO REMOVE PARAMETER
-----05/31/85-15.50-----

```

Display Current CPU's Memory

Syntax: "M <start_addr>"
Or: " <start_addr>"

This command causes the currently selected CPU's memory to be displayed in sequential lines containing the following information:

- the address with corresponding logical partition number if applicable,
- 8 double spaced words of memory in hexadecimal format, and
- a printable character representation of the 8 words of memory (unprintable characters are replaced with periods).

The "start_addr" parameter specifies the starting address from which memory will be displayed. Note that if HAZEL is the currently selected CPU, the specified start address will be added to the base address set by the the "B=" global command. If "M" is entered alone, memory will be displayed beginning at the location last displayed.

IBM Internal Use Only

Display the Currently Selected CPU's Registers

Syntax: "R <XR_page>"

This command causes the contents of the following registers to be displayed for the currently selected CPU:

- 8 levels of local storage registers (LSRs) (16 registers per level),
- 32 external registers, and
- 8 levels of program status words (PSWs).

Three words per level are displayed in level order running from left to right, top to bottom. These 3 words contain the following information:

1. the instruction address register (IAR),
2. the condition codes and LSPRS, and
3. condensed ILDE data (each of the 4 digits represent the low order nibble in the corresponding byte of data).

If Debug is in AP mode, only 6 levels of LSRs are displayed, and condensed ILDE data is not displayed. The optional "XR_page" parameter is a number between 0 and 7 indicating which page of the AP's external registers are to be displayed. This parameter is ignored if Debug is in HAZEL mode.

Dump Memory to S/1

Syntax: "DU"

This command is used to dump either AP or HAZEL memory to diskette. To initiate the DSC to S/1 dump, perform the following steps in sequence:

1. Select the desired CPU (HAZEL or AP) and partition using the appropriate global commands,
2. Enter the DU command,
3. Observe the AP light pattern - lights 0 to 3 should start flashing,
4. Determine the line number of the DSC which is required by the DTD program,
5. Insert the EDXDMP diskette into S/1 diskette drive, and
6. Load the DTD program and enter requested information.

Hit ENTER if the dump abends, or to terminate the dump while it is in progress. The ENTER key disarms the dump and returns control to Debug. Refer to the DSC to S/1 DUMP manual for further information.

Display System Queues

Syntax: "SQ"

This command displays information pertaining to the following system queues:

- Active Task
- Ready Queue
- Timer Queue

IBM Internal Use Only

- CCB - ECB 1
- CCB - QCB 1
- CCB - ECB 2
- CCB - QCB 2
- Cancel Queue

The following information is displayed for each queue member:

1. the partition key,
2. the TCB address,
3. the contents of #1 and #2, and
4. the contents of 16 local storage registers.

If a queue is empty, the queue title will appear without any associated information. Display will halt after the display of each queue's information. To view the next queue's information, the user must hit the enter key. Queue information will eventually be redisplayed if the user continues to hit the enter key. The user may exit system queue display at any time by entering a new command.

Return to Previous Primary Subsystem

Syntax: "X"

This command is used to return the user to the previous primary subsystem. Up to eight previous subsystems may be accessed in succession by repeatedly specifying the "X" command. If the user attempts to back up more than eight levels, the primary commands menu will be displayed.

Display Global Settings

Syntax: "G="

This command is used to display the global commands menu and the current settings of the global parameters. The following is an example of the Global Commands Menu:

GLOBAL COMMANDS		CURRENT SETTINGS
HA/AP	- HAZEL/AP MODE	HA
CP (KEY)	- CHANGE PARTITION (1 TO 3)	0001
B= (ADDR)	- SET BASE ADDRESS	0000
DM (ADDR)	- DISPLAY MEMORY ON EDX STOP	0000
TS/TL	- SHORT(256)/LONG(24,576) EDX TRACE	TS
PM/SM	- PICTURE/SCROLL MODE	PM
KS/FS	- KEEP/FREE DEBUG SCREEN ON STOP	FS
R?/R.	- DO/DO NOT DISPLAY UNCHANGED REGS ON STOP	R.
L (COUNT)	- SET LINE COUNT	0004
W (COUNT)	- SET WORD COUNT	0032
SP (LINES)	- SET DISPLAY SPACING (0 TO 7)	0000
SK (LINES)	- SKIP LINE(S) (-7 TO 7)	

IBM Internal Use Only

Quit Debug

Syntax: "Q"

This command allows the user to quit Debug, and return to the application. If any JIB or EDX break points remain outstanding, Debug will continue to monitor execution, and will reactivate itself when a break point is reached. Similarly, JIB or EDX tracing will continue until the user re-enters Debug and explicitly requests that tracing be stopped.

Stop Jib

Syntax:

```
"SJ      <stop_addr cnt iter |  
          stop_addr cnt |  
          stop_addr |      >"
```

This command is used to stop HAZEL's execution of JIB code at a designated point. The following parameters specify where execution is to be halted:

stop_addr specifies the address at which execution is to be halted.

cnt specifies the maximum number of statements that will be executed following the statement at which HAZEL is currently stopped. When specified alone, this parameter defines the number of instruction that are to be executed before execution is halted. When specified in conjunction with a stop address, the count acts as a backup for situations where the execution path is uncertain. Valid values range from 1 to 256. This parameter is ignored if HAZEL is not stopped at the time of specification.

iter specifies the number of times the specified instruction is to be executed before HAZEL is stopped. Valid values range from 1 to 256.

Parameters may be omitted by placing zeros in the appropriate positions within the parameter list. If only zero(s) are specified, any outstanding break points are removed, the application screen is restored, and HAZEL is allowed to execute freely. If the "SJ" command is issued without any associated parameters, HAZEL is stopped on the JIB instruction currently being executed. A new break point may be specified at any time regardless of whether or not the previously specified break point has been reached. Since only one break point can be outstanding at a time, specification of a new break point will automatically remove the old break point. If HAZEL is stopped, the user may single step the code by hitting the enter key.

This function is not capable of stopping on, or single stepping across a spi or a plex. Therefore, explicit break points must be placed before and after spis and plexes if the user wants to debug code involving such statements. Stopping on addresses which have interrupts disabled may only be done by first single stepping across the interrupt disable instruction. Similarly, before subsequently stopping on addresses which have interrupts enabled, the user must first single step across the interrupt enable instruction. Note that the appropriate partition must be established using the "CPF" global command when setting break points at addresses above 7FFF.

As each break point is reached, the following information is displayed:

- the partition key and address at which HAZEL is stopped,
- the instruction found at the stop address,
- 16 local storage registers,
- the condition code flags (E=equal, Z=zero, C=carry),

IBM Internal Use Only

- the level, and
- the chapter and LSPRs.

A line of titles describing this information may be displayed by issuing the "I" command. If HAZEL is not stopped, this command is ignored. If the global command "R." has been issued, then only modified local storage register pairs are displayed following the first stop. Those which have not been modified, are shown to contain periods "." (refer to Global Commands). While HAZEL is stopped on an address, the user may execute any of the Debug commands. If the user exits "stop JIB" mode by executing a new primary command, the user must issue either the "X" (previous primary subsystem), or the "RD" (redisplay) command to reenter "stop JIB" mode. Upon reentering "stop JIB" mode, Debug will display the current "stop JIB" information, and then prompt the user for a "stop JIB" secondary command (refer to Stop Secondary Commands).

By default, the application screen will be restored upon the issuing of a "SJ" command. If no screen I/O is to take place before the break point is reached, the user may issue the "KS" (keep screen) command, causing the Debug screen to be kept (ie. as opposed to having the application screen restored). The user can again have the application screen restored between break points by issuing the "FS" (free screen) command.

stop EDX

Syntax:

```
"SE      <stop_addr tcb_addr cnt iter |
          stop_addr tcb_addr cnt |
          stop_addr tcb_addr |
          stop_addr | >"
```

This command is used to stop HAZEL's execution of EDX code at a designated point. The following parameters specify where execution is to be halted:

- stop_addr** specifies the address at which execution is to be halted.
- tcb_addr** specifies the task control block which is to be active when execution is halted on the specified address.
- cnt** specifies the maximum number of statements that will be executed following the statement at which HAZEL is currently stopped. When specified alone, this parameter defines the number of instruction that are to be executed before execution is halted. When specified in conjunction with a stop address, the count acts as a backup for situations where the execution path is uncertain. Valid values range from 1 to 256. This parameter is ignored if HAZEL is not stopped at the time of specification.
- iter** specifies the number of times the specified instruction is to be executed before HAZEL is stopped. Valid values range from 1 to 256.

Parameters may be omitted by placing zeros in the appropriate positions within the parameter list. If only zero(s) are specified, any outstanding break points are removed, the application screen is restored, and HAZEL is allowed to execute freely. If the "SE" command is issued without any associated parameters, HAZEL is stopped on the EDX instruction currently being executed. A new break point may be specified at any time regardless of whether or not the previously specified break point has been reached. Since only one break point can be outstanding at a time, specification of a new break point will automatically remove the old break point. If HAZEL is stopped, the user may single step the code by hitting enter.

As each break point is reached, the following information is displayed:

- the partition key and address at which HAZEL is stopped,
- the contents of the index registers, and

IBM Internal Use Only

- 8 words of memory specified by the "DM" global command.

A line of titles describing this information may be displayed by issuing the command "T". If HAZEL is not stopped, this command is ignored. While HAZEL is stopped on an address, the user may execute any of the Debug commands. While HAZEL is stopped on an address, the user may execute any of the Debug commands. If the user exits "stop EDX" mode by executing a new primary command, the user must issue either the "X" (previous primary subsystem), or the "RD" (redisplay) command to reenter "stop EDX" mode. Upon reentering "stop EDX" mode, Debug will display the current "stop EDX" information, and then prompt the user for a "stop EDX" secondary command (refer to Stop Secondary Commands).

By default, the application screen will be restored upon the issuing of a "SE" command. If no screen I/O is to take place before the break point is reached, the user may issue the "KS" (keep screen) command, causing the Debug screen to be kept (ie. as opposed to having the application screen restored). The user can again have the application screen restored between break points by issuing the "FS" (free screen) command.

Stop Titles

Syntax: "T"

This command is used to redisplay the stop information title associated with the current EDX or JIB stop (refer to the Stop EDX and Stop JIB commands). If the HAZEL is not stopped, this command will be ignored.

Redisplay Stop Information

Syntax: "RD"

This command is used to reenter "stop JIB" or "stop EDX" mode depending on which type of code is currently stopped. The stop information associated with the current EDX or JIB stop (refer to the Stop EDX and Stop JIB commands) is displayed upon specification of this command. If the HAZEL is not stopped, this command will be ignored.

HAZEL Soft IPL

Syntax: "IP"

AP-CSS first reinitializes its CSS channels and its internal control blocks. Then, it goes through the HAZEL ipl sequence. Depending on what is the ipl device HAZEL will be ipl'd from EPROM card, AP-CSS or AP-disk. The name of the nucleus loaded by AP-CSS is NC3 and by the AP-disk is \$EDXIPL.

Trace EDX

Syntax: "TE" <start_addr tcb_addr cnt iter stop_addr |
 start_addr tcb_addr cnt iter |
 start_addr tcb_addr cnt |
 start_addr tcb_addr |
 start_addr | ">"

This command is used to trace a designated section of EDX code. The following parameters specify the section of code to be traced:

start_addr specifies the address at which tracing is to begin.

IBM Internal Use Only

- tcb_addr** specifies the task control block which is to be active when tracing begins.
- cnt** This parameter defines the number of statements that will be traced following the specified start address. If the start_addr parameter is not specified, this parameter defines the number of statements that will be traced following the statement at which the HAZEL is currently stopped. Valid values range from 1 to 256.
- iter** specifies the number of times the instruction at the specified start address is to be executed before tracing is to begin. Valid values range from 1 to 256.
- stop_addr** specifies the address at which tracing is to end.

Parameters may be omitted by placing zeros in the appropriate positions within the parameter list. If only zero(s) are specified, tracing is terminated, and HAZEL is allowed to execute freely. If no associated parameters are specified, EDX code is traced indefinitely beginning with the instruction currently being executed. New trace parameters may be specified at any time.

EDX Trace Size

Global commands TS (SMALL) and TL (LARGE) may be used to select EDX trace buffer size:

1. SMALL buffer - 256 words in the nucleus.
2. LARGE buffer - 24,576 words in the user memory (A000 to FFFF partition 2).

The default trace buffer size is SMALL.

View EDX Trace

Syntax: "ET"

This command is used to view the contents of the EDX trace table. The EDX trace is displayed in three or more columns: the first column contains a line number, the second column contains a TCB address, and subsequent columns contain "post", "chain", "TCB", and instruction address information. A post is indicated by:

- the letters "P__",
- the level, and
- the associated ECB address.

A chain is indicated by:

- the letters "C__",
- the level, and
- the associated TCB address.

The EDX trace will also display which instructions were executed by a given task. The TCB address is displayed in the second column, while associated instruction addresses are displayed sequentially from left to right in the next 7 columns. The user may scroll through the EDX trace as he would scroll through memory. Display of the EDX trace begins with the oldest part, whether or not it was generated by the most recent "IE" command. Scrolling in either direction will eventually wrap around and redisplay information that has already been viewed.

IBM Internal Use Only

Memory Secondary Commands

Memory Commands Menu

MEMORY COMMANDS

P (PATCH1)---(PATCH10)	- PATCH AT CURRENT LOCATION
P= (PATCH1)---(PATCH10)	- PATCH FOLLOWING LAST PATCH
PA (ADDR)(PATCH1)---(PATCH9)	- PATCH AT GIVEN ADDRESS
PR (PATCH)(COUNT)	- PATCH REPEAT
I (HEXCOUNT)	- INDEX UP/DOWN
S (WORD)(COUNT)	- SEARCH FOR WORD
T (FROM)(TO)(DEST)(DESPARTN)	- TRANSFER CURRENT CPU'S MEMORY
	- SCROLL MEMORY
RS	- REVERSE SCROLL DIRECTION
X	- PREVIOUS PRIMARY SUBSYSTEM
=X	- PRIMARY COMMAND MENU
?	- AVAILABLE COMMANDS

Patch Memory at Currently Displayed Location

Syntax:

"P <patch1 | patch1 patch2 | patch1 patch2 patch3 ...>"

This command is used to patch memory starting with the left, uppermost word in the block of memory last displayed. Up to 10 sequential words may be patched at a time. Intervening words which are not to be patched may be left untouched by inserting commas in the appropriate positions within the parameter list. For example:

P AAAA,BBBB,CCCC,DDDD will patch every other word beginning with the left, uppermost word in the block of memory last displayed.

Up to 10 words over a range of 16 words may be patched when using intervening commas.

Patch Memory at Following Last Patch

Syntax:

"P= <patch1 | patch1 patch2 | patch1 patch2 patch3 ...>"

This command is used to patch memory directly following the last patch made. Up to 10 sequential words may be patched at a time. Intervening words which are not to be patched may be left untouched by inserting commas in the appropriate positions within the parameter list (refer the "P" command).

Up to 10 words over a range of 16 words may be patched when using intervening commas.

Patch Memory at a Specified Address

Syntax: "PA address <patch1 | patch1 patch2 ...>"

This command is used to patch the memory of the currently selected CPU starting at the address specified by the first parameter (plus a base address set by the "B=" command if patching HAZEL's memory). Up to 9

IBM Internal Use Only

sequential words may be patched at a time. Intervening words which are not to be patched may be left untouched by inserting commas in the appropriate positions within the parameter list.

Up to 9 words over a range of 15 words may be patched when using intervening commas.

Patch Repeat Word in Memory

Syntax: "PR <pattern WORD_CNT | pattern | >"

This command causes the memory of the currently specified CPU to be changed according to the parameter(s) specified:

pattern specifies the word of data that is to be placed in memory (the default is x'0000').

WORD_CNT is a decimal number specifying how many words are to be replaced with the specified or default word (the default is a count of 1 word).

Index Through Memory

Syntax: "I <offset>"

This command allows the user to index through memory a specified number of words relative to the memory location currently displayed. The specified offset is assumed to be in hexadecimal. By prefixing the offset parameter with a minus sign "-", previous addresses may be referenced.

Search for Word in Memory

Syntax: "S <pattern WORD_CNT | pattern | >"

This command searches memory according to the the parameter(s) entered:

pattern specifies the word of data which is to be located (the default is x'0000').

WORD_CNT is a decimal number specifying the number of words, starting from the left uppermost corner of the block of memory last displayed, that are to be scanned for the specified pattern.

To continue searching following the last found pattern, reissue the "S" command without any associated parameters.

Transfer Memory

Syntax:

"T start_addr end_addr dest_addr <dest_partn>"

This command is used to copy a specified block of memory to another location either within one CPU or across CPUs. The source CPU, from which memory is copied, is defined by the current operating mode (either HAZEL or AP, global commands). If HAZEL is the source CPU then the current partition setting defines the source partition from which to copy. The following parameter descriptions explain how this command is used:

start_addr specifies the source start address of the block of memory to be copied.

IBM Internal Use Only

end_addr specifies the last address of the source block of memory.

dest_addr specifies the starting address of the destination memory location.

dest_partn if this parameter is specified, then HAZEL is the destination CPU.

Scroll through Memory

Syntax: " "

This command causes memory display to scroll in the direction currently specified.

Reverse Scrolling Direction

Syntax: "RS"

This command causes the display of the currently selected CPU's memory to reverse scrolling direction.

Register Secondary Commands

Register Commands Menu

REGISTER COMMANDS

S (CHAPTER)(BLOCK)	- SELECT REGISTERS
T	- REGISTER TITLE DISPLAY
PR (LVL0-7)(REG0-F)(PAT1)(PAT2)	- PATCH HAZEL LSRS
PX (XREG0-1F)(PAT1)(PAT2)	- PATCH HAZEL XREGS
PX (XREG0-1F)(DATA)	- PATCH AP XREGS
X	- PREVIOUS PRIMARY SUBSYSTEM
=X	- PRIMARY COMMAND MENU
?	- AVAILABLE COMMANDS

Select Registers by Chapter and Block

Syntax: "S <chapter block | chapter | >"

This command is used to display HAZEL registers by chapter/block convention. The following parameters specify what information is to be displayed:

chapter specifies the chapter (0->7) from which registers are to be displayed (The default is chapters 0 through 3. If the user enters another "S" command, chapters 4 through 7 are displayed).

block specifies the pair of blocks within the specified chapter that are to be displayed (The default is all of the blocks: 0 through 7).

If an even numbered block is specified, that and the following block are displayed. If an odd numbered block is specified, that and the previous block are displayed.

IBM Internal Use Only

Display Register Titles

Syntax: "T "

This command displays a line of titles for a chapter/block register display.

Patch Local Storage Registers

Syntax: "PR level lsr_no <patch1 patch2 | patch1>"

This command is used to patch local storage registers individually, or in even-odd pairs. The following parameter descriptions explain how this command is used:

lvl_no lsr_no these two parameters specify which register is to be patched: the first parameter specifies the level (0->7), and the second parameter specifies the register (0->F).

patch1 specifies the patch to be made to the specified register.

patch2 If this parameter is entered and an even numbered register has been specified, this specifies the patch to be made to the following odd numbered register.

Patch HAZEL External Registers

Syntax: "PX xreg_no <patch1 patch2 | patch1>"

This command is used to patch HAZEL's external registers individually, or in even-odd pairs. The following parameter descriptions explain how this command is used:

xreg_no specifies which external register (0->1F) is to be patched.

patch1 specifies the patch to be made to the specified external register.

patch2 If this parameter is entered and an even numbered external register has been specified, this specifies the patch to be made to the following odd numbered external register.

Patch AP External Registers

Syntax: "PX xreg_no <data>"

This command is used to patch AP's external registers one at a time. The following parameter descriptions explain how this command is used:

xreg_no specifies which external register (0->1F) is to be patched.

data specifies the patch to be made to the specified external register.

IBM Internal Use Only

Stop Secondary Commands

Stop Commands Menu

STOP COMMANDS

- * GE (ADDR) - SINGLE STEP JIB/EDX
- * PI (#1)(#2) - GOTO EDX INSTRUCTION ADDRESS
- = (ADDR) - PATCH EDX INDEX REGISTERS
- X - DISPLAY MEMORY
- =X - PREVIOUS PRIMARY SUBSYSTEM
- ? - PRIMARY COMMAND MENU
- ? - AVAILABLE COMMANDS

* - PERTAIN ONLY TO EDX STOPS

ENTER CMD:

Single Step JIB/EDX

Syntax: " "

This command is used to single step JIB or EDX code depending on which type of code is currently stopped.

IBM Internal Use Only

Go to User Specified Address

Syntax: "GE <addr>"

This command is used to change the HAZEL's instruction address register (IAR) to the specified address, and to halt execution when this address is reached.

Patch EDX Index Registers

Syntax: "PI <patch1 patch2 | patch1 | , patch2>"

This command is used to patch one or both of the EDX index registers. Note that if only #2 is to be patched, the desired patch must be prefixed by a comma so as to leave #1 untouched.

Display Memory

Syntax: "= <addr>"

This command is used to display a specified or default area of memory. If an address is not specified, the memory last displayed will be redisplayed. Note that global commands pertaining to memory display remain active. This command only displays an area of memory. To perform other memory functions, the user must enter "memory" mode by executing the "M" primary command.

IBM Internal Use Only

APPENDIX D. EPROM BURN FACILITY

The EPROM burn facility is a specialized DSS package which provides the support needed to program (burn) data on the two EPROM card types used to IPL the DSC. These two cards are as follows :

1. EPROM I card P/N 8912389 (16K words of EPROM)
2. EPROM II card P/N 8692032 (64K words of EPROM)

The burn facility also provides the user with the ability to program the EPROM module sets required on the Hazel Attached Processor (AP) card, P/N 6229533.

The "AP" EPROMs are burned on the 64k EPROM II card, then the EPROM modules are removed and reinstalled in the sockets provided on the "AP" card.

IBM Internal Use Only

EPROM BURNER UTILITY (BURN)

This utility is used in conjunction with the DSC IV hardware package designed for programming "Burning" the EPROM card types described. The following functions are provided:

- Program "BURN" EPROM I/EPROM II cards, and i2764A EPROM module sets to be installed on the Hazel Attached Processor (P/N 6229533) cards.
- Copy EPROM card(s).
- Compare EPROM card to memory.
- Burn EPROM card(s) with EDX/J Nucleus and application programs for standalone system operation.
- Provision for handling (2) 16K EPROM cards in any of the available options for data lengths up to 32K words. Operations will be done one card at a time-- the operator is prompted to remove the initial card and install the second 16K card to continue the operation.
- Complete Program control over the auxiliary PROGRAM VOLTAGE power supply-- eliminating the need for the operator to adjust voltages prior to EPROM card removal, or at any other operational phase. A special voltage regulator card will be needed to accomplish this. If unavailable, the program voltage must be set prior to the BURN/VERIFY stage, and lowered back to 5 vdc for all other operations. Caution must be exercised to NOT exceed the program voltage setting. Otherwise irreputable damage to the EPROM devices will result.

IBM Internal Use Only

The "Burn System" requires an auxiliary power supply to be attached. This supply will provide the program voltages for the two different types of EPROM cards, +25 and +12.5 vdc⁵ for the EPROM I and II cards respectively. AC power to this supply is program controlled via SOLID STATE RELAY in the burner system.

```
*****
*   H A Z E L   E P R O M   B U R N   P R O G R A M   *
*****mm/dd/yy-hh.mm**

1 ==> FOR 16K EPROM CARD
2 .... FOR 64K EPROM CARD
3 .... FOR ATTACHED PROCESSOR EPROMS
ENTER TYPE ==>
```

Figure 9. EPROM burner initial display.

Enter the EPROM type you will be working with or, just press ENTER if the arrows are pointing to the desired option. The arrows point to the next default option, thus providing ENTER key operation for normal program operation. This utility will predict the next logical option to help minimize key entry.

```
*****
*   H A Z E L   E P R O M   B U R N   P R O G R A M   *
*****mm/dd/yy-hh.mm**

1 ==> NUCLEUS LOAD
2 .... PROGRAM LOAD
3 .... BURN EPROM CARD W/ERASE VERIFY
4 .... BURN EPROM CARD W/O ERASE VERIFY
5 .... DUMP EPROM CARD INTO MEMORY
6 .... COMPARE EPROM CARD CONTENTS TO MEMORY
7 .... CHANGE EPROM CARD TYPE
E .... EXIT EPROM BURN PROGRAM
ENTER TYPE ==>
```

Figure 10. EPROM burner utility main option menu.

1 NUCLEUS LOAD: This option allows the user to load data via "NATIVE DISK" or Series/1 "HOST" link into DSC IV memory. When this option is selected, the "RUN OPTIONS" menu will be displayed thus allowing the operator to select the load source (HOST/DISK). After the load source has been selected, the name of the data file is "entered" and the operation is initiated. Entering 'N' or 'NO' at the prompt for "FILE NAME" will exit this option without loading any data.

2 PROGRAM LOAD: This option is executed similar to option 1, except, the data to be loaded will be an EDX/J PROGRAM LOAD MODULE. This option completely emulates the EDX/J nucleus functions (LOAD/LOADH) using the EPROM card as the target memory. NOTE: This function will abort unless a compatible EDX/J nucleus has been previously loaded via option 1.

Testing is recommended prior to burning the programs on EPROM. An equivalent method of testing the nucleus/programs without first burning

⁵ The required program voltage is version dependent of the EPROM device used. Consult the device hardware specification.

IBM Internal Use Only

them on EPROM is possible. This may be done by performing an IPL on a compatible system with the EDX/J nucleus to be used. Then load all of the selected programs but do not attach them. If one of the programs is to be attached automatically after the IPL, attach that program using the '\$\$' attention function but only after all programs have been stored in memory. The result of this procedure will be as if the nucleus and programs were IPLed from EPROM.

The user has the option of selecting one (1) program to be "attached" at IPL time. TO ATTACH A PROGRAM, select option "A" on the "RUN OPTIONS" menu BEFORE initiating the load for the program to be attached. This operation may be done to as many programs desired without any ill effects, but because of provision for only one program attach, the function will be set for the last selected program to be attached. Entering 'N' or 'NO' at the prompt for "FILE NAME" will exit this option without loading any data.

3 BURN EPROM CARD W/ERASE VERIFY: This option will first verify that all addresses to be burned on the EPROM card are blank, then allows the EPROM card to be burned. This option uses the same "RUN VALUES"⁶ previously entered, and allows the change of those parameters if desired. NOTE: The program voltage must be present during this operation.

4 BURN EPROM CARD W/O ERASE VERIFY: This option is used the same as option 3 except the program is set to burn only. The EPROM addresses to be burned are not verified to be blank before the burn. This is useful when burning spare locations on a previously burned card. NOTE: The program voltage must be present during this operation.

5 DUMP EPROM CARD INTO MEMORY: This option will dump the user specified contents of a selected EPROM card into memory.

TO COPY AN EPROM CARD, the "MASTER" EPROM card is plugged into the proper EPROM card slot, its contents are dumped into memory, then the blank EPROM card is inserted in its place. Option 3 or 4 is then selected and the data that was dumped into memory is burned on the blank card.

6 COMPARE EPROM CARD CONTENTS TO MEMORY: This option is used mainly after a burn is completed. A compare of the "BURN DATA" (in memory) with the EPROM card data is necessary after a burn to verify the contents of the EPROM card. The data is verified as each location is burned, but only at the PROGRAM VOLTAGE⁵. It is necessary to compare the contents of the EPROM card while at the READ VOLTAGE⁷ which is also the voltage that will be applied during normal card operation.

7 CHANGE EPROM CARD TYPE: This option provides the method of changing the EPROM card type being used. It is necessary to input the proper EPROM type to obtain the correct results from this utility.

E EXIT EPROM BURN PROGRAM: This option is used to detach the burn program. The program may be restarted with the '\$\$' attention function.

RUN OPTIONS MENU: The following figure is displayed before and during each of the main options, thus providing the user with a complete display of all system parameters.

⁶ Hazel Start Address, Key, EPROM Start Address, and Data Length.

⁷ EPROM voltage (Vpp = +5V) needed for normal "READ" operation.

IBM Internal Use Only

```
*****
*   H A Z E L   E P R O M   B U R N   P R O G R A M   *
*****mm/dd/yy-hh.mm**

HAZEL(ADDR KEY) : 8000-1          MACHINE TYPE   : 16K HAZEL
EPROM ADDRESS   : 0000          FILE NAME      :
LENGTH OF DATA : 00004000      ATTACH PROGRAM : NO
LRC GENERATION   : OFF          COPIES         : 1
LOAD SOURCE      : DISK         CURRENT FUNCTION: NUC LOAD
=====

C => CHANGE (HAZEL_ADDR KEY EPROM_ADDR DATA_LENGTH)
MC => GENERATE MULTIPLE COPIES OF NUCLEUS
S => CHANGE LOAD SOURCE (HOST/DISK)
A => SET/RESET IPL PROGRAM ATTACH
L => TURN ON/OFF LRC GENERATION
X => RETURN TO MAIN MENU
ENTER TO START =>
```

Figure 11. EPROM burner "Run Options" menu.

C => CHANGE (HAZEL_ADDR KEY EPROM_ADDR DATA_LENGTH): Selecting this option allows the user to change all or any one of the "RUN VALUES". The "RUN VALUES" are ; HAZEL START ADDR, KEY, EPROM START ADDR, and LENGTH OF DATA respectively. If desired, each parameter may be changed (within allowable limits). Advanced key input is excepted for all 4 parameters*.

MC => GENERATE MULTIPLE COPIES OF NUCLEUS This function is mainly used for burning multiple sets of Hazel Attached Processor (AP) EPROMs. Select this option after all "RUN VALUE"* adjustments have been completed. The maximum number of copies is calculated based on the "RUN VALUES" and will be displayed with a prompt for the desired number of copies. The utility will default to the maximum number of copies unless specified otherwise. After the "BURN" operation, the copies may be separated into individual module groups. The operator must keep track of the original file size and EPROM address to determine how to separate the modules into the individual copies. EPROM card data organization figures are available later in this section which provide the needed module placement information.

NOTE : A copy is defined by the minimum number of EPROM module groups needed to contain the data defined in the "RUN VALUES". An EPROM module group is composed of two (2) EPROM modules which are of the same address range and together form a complete data word.

S => CHANGE LOAD SOURCE (HOST/DISK): On selection, the current load source displayed is replaced by the alternate source-- HOST is replaced by DISK etc. This parameter sets the data origin used by the Nucleus and Program Load options.

A => SET/RESET IPL PROGRAM ATTACH: Enable program attach feature for the NEXT program loaded. The EDX/J nucleus contains the provision to attach one (1) program after its initialization procedures have completed. This option provides the "EPROM NUCLEUS"* with the address of the next loaded program so that it may be attached automatically after system IPL.

L => TURN ON/OFF LRC GENERATION: This option will enable/disable the generation of an LRC. If enabled, a Longitudinal Redundancy Code (LRC) is calculated and burned in a specific location on the EPROM card. This

* All parameters may be entered on the same line separated by spaces or commas before pressing ENTER.

* The EDX/J nucleus to be burned on the EPROM card.

IBM Internal Use Only

feature is only used when burning a DSC nucleus. If configured, the LRC will be checked when the DSC is going through its initial bring-up diagnostics. This code is used to help verify that the EPROM card data, when loaded into the DSC memory, did so without error. The LRC is generated prior to the BURN operation.

The specific locations where the LRC is stored is as follows:

- DSC II & III (HEX) ADDRESS => 3FFF
- DSC IV (HEX) ADDRESS => 7FFF
- Hazel Attached Processor (HEX) ADDRESS => 3FFF

X => RETURN TO MAIN MENU: This selection will exit the current routine before it is executed.

Bits (0- 7) Address (0800- 0FFF) (2) .	Bits (0- 7) Address (0- 07FF) (1) .	Bits (0- 7) Address (1800- 1FFF) (4) .	Bits (0- 7) Address (1000- 17FF) (3) .	Bits (0- 7) Address (2800- 2FFF) (6) .	Bits (0- 7) Address (2000- 27FF) (5) .	Bits (0- 7) Address (3800- 3FFF) (8) .	Bits (0- 7) Address (3000- 37FF) (7) .
Bits (8-15) Address (0- 07FF) (1) .	Bits (8-15) Address (0800- 0FFF) (2) .	Bits (8-15) Address (1000- 17FF) (3) .	Bits (8-15) Address (1800- 1FFF) (4) .	Bits (8-15) Address (2000- 27FF) (5) .	Bits (8-15) Address (2800- 2FFF) (6) .	Bits (8-15) Address (3000- 37FF) (7) .	Bits (8-15) Address (3800- 3FFF) (8) .

Figure 12. EPROM I card p/n 8912389 data organization.

IBM Internal Use Only

Bits (0- 7) Address (0- 1FFF) (1) .	Bits (0- 7) Address (2000- 3FFF) (2) .	Bits (0- 7) Address (4000- 5FFF) (3) .	Bits (0- 7) Address (6000- 7FFF) (4) .	Bits (0- 7) Address (8000- 9FFF) (5) .	Bits (0- 7) Address (A000- BFFF) (6) .	Bits (0- 7) Address (C000- DFFF) (7) .	Bits (0- 7) Address (E000- FFFF) (8) .
Bits (8-15) Address (0- 1FFF) (1) .	Bits (8-15) Address (2000- 3FFF) (2) .	Bits (8-15) Address (4000- 5FFF) (3) .	Bits (8-15) Address (6000- 7FFF) (4) .	Bits (8-15) Address (8000- 9FFF) (5) .	Bits (8-15) Address (A000- BFFF) (6) .	Bits (8-15) Address (C000- DFFF) (7) .	Bits (8-15) Address (E000- FFFF) (8) .

Figure 13. EPROM II card p/n 8692032 data organization.

IBM Internal Use Only

EPROM MODULE PLACEMENT (Attached Processor): The following two figures illustrate the EPROM module placement on the EPROM II (burn card) to their respective locations on the AP card.

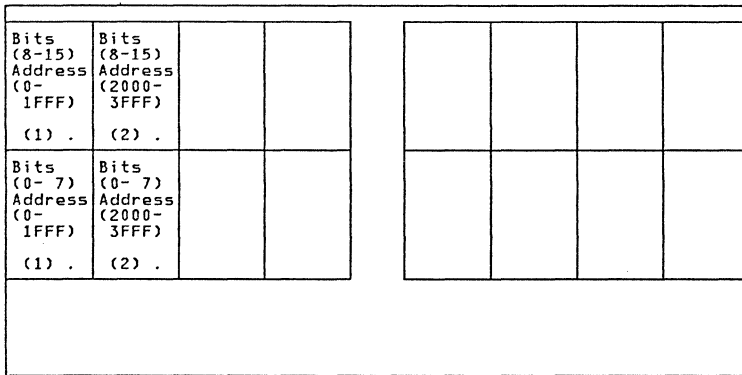


Figure 14. EPROM II card. Locations and data organization for At-
tached Processor (AP) EPROM modules.

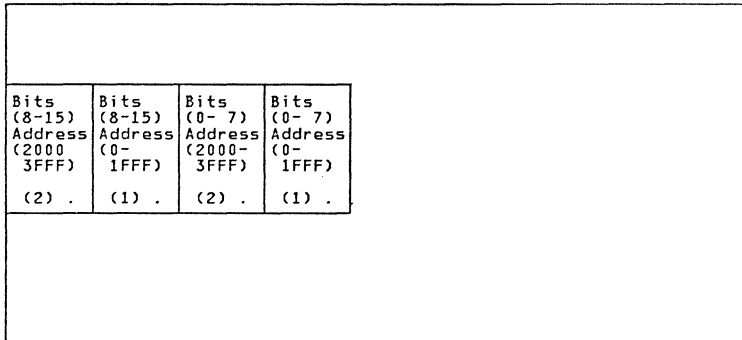


Figure 15. Attached Processor card. EPROM module organization.

IBM Internal Use Only

APPENDIX E. COMMUNICATION ERROR CODES

This section contains a list of the communication error codes used with the DSC to Series/1 (DIS) communication, and DSC/Disk support packages. The following diagram illustrates the format of the error code.

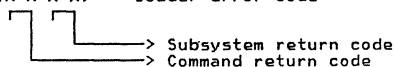
PROGRAM LOAD ERRORS

EXAMPLE:

LOAD ERROR XXXX

WHERE:

(X X X X) <= Loader error code



The load error code is comprised of a return code from the higher level command (LOADH/LOAD), and the return code from the subsystem (SEND/RECEIVE & WRITE/READ) used by that command.

```

10XX      Program not found                (Native Disk)
20XX      Send Failed. (Initial request/ File Query)
30XX      Receive Failed. (Response to initial request/ File Parameters)
40XX      Invalid Program name             ( Host response )
50XX      Program not found                ( Host response )
60XX      Not a tool program               ( Host response )
70XX      Disk error                      ( Host response )
80XX      Loader2 program unloadable      ( Host response )
90XX      Not enough memory for load
A0XX      Send failed                     (Data Request)
B0XX      Receive/Read failed             (Data transfer)
C0XX      Record number mismatch
D000      Program level error
E000      DIS Interrupt bit conflict.     (Interrupt already defined)

```

IBM Internal Use Only

COMMUNICATION SUBSYSTEM RETURN CODES (SEND/RECEIVE)

HARDWARE ERRORS ON TRANSMIT

2200 - DIS error on transmit

ERRORS FROM SEND/RECEIVE SUPPORT

3000 - Send command or receive port in use	S/R
3100 - Port 0 not in Big Table	S/R
3200 - OCB pointer missing from command	
3400 - TP facilities not open by program	S/R
3500 - Port number out of range	S/R
3600 - Word count out of range	S/R
3700 - Message priority out of range	S
3800 - Invalid destination node number	
4000 - Receive wait timeout	R
4100 - Receive message too big	R
4200 - No work exit code (set in TCB)	R
4300 - Communication Line not enabled	

PERMANENT ERRORS RECEIVED FROM OTHER SIDE

5000 - Destination not found
5100 - Destination marked not loadable
5200 - Receive buffer is too small
5300 - No free buffer available
5500 - No receive buffer supplied by RECEIVE command

RETRIABLE TP ERRORS SET BY ORIGINATOR

8600 - Timeout on ACK from other side
8700 - TX buffer not emptied
8800 - Error on ACK from other side

RETRIABLE TP ERRORS RECEIVED FROM OTHER SIDE

E000 - Base code for RS232 status
E200 - Framing error
E400 - Parity error
E900 - LRC/CRC error
EB00 - Complete blast reset received
EC00 - Invalid count field received
ED00 - DIS error on receive
EF00 - Blast reset fragment received
FF00 - Initial bid from other side, or other side timed out.

DISK ERROR RETURN CODES (NATIVE DISK SUPPORT)

0000 - Drive ready/Dataset found
0001 - Drive not ready/No disk
0002 - Seek error
0003 - Read/Write error
0004 - Hardfile Format error
0005 - CTS error
0006 - Timeout
0007 - More than two bad tracks
0008 - Soft error reporting
00BF - Disk errors from Adapter
00FF - Dataset not found

IBM Internal Use Only

APPENDIX F. DIS ERROR CODES

DIS error codes are used to describe a failing condition on the DIS Channel. These error codes are available to the application programmer for display if desired, thus they may or may not be displayed depending on the way the application programmer handles these errors. They will always be stored in the DIS error trace table located within the EDX/J nucleus. These may be viewed in memory starting at the address of the system label \$DISTRAC. You will need the nucleus listing for your particular system to do this.

DIS errors 1 - F (HEX) hang the machine on level 3.

The Xs will contain the command tag used when the fail occurred. If they are zero, then the program has filtered the command tag out and left only the error code, or a "select" operation was issued. The command tag (CTC) and error will be displayed in the DIS trace table.

- XX00 (HEX) - Command successful.
- XX01 (HEX) - Hot return tag on data transfer. This error is posted when the DIS Channel detects the 'RETURN' tag active while attempting to transfer data to/from one of the macrofunction cards.
- XX02 (HEX) - Hot return tag on select sequence. This error is posted when the DIS Channel detects the 'RETURN' tag active while attempting to address a macrofunction card.
- XX03 (HEX) - Hot Return during poll operation. This error is posted when the 'RETURN' tag is detected active when attempting to POLL for interrupt acknowledge.
- XX04 (HEX) - Hot Interrupt, Poll not acknowledged. This error is posted when 'INTERRUPT REQUEST' is active, and no acknowledge was received from any of the BICs in the BIC POLL LIST, or an interrupt could not be reset.
- XX05 (HEX) - Hot return detected on reading BIC reference register.
- XX06 (HEX) - Machine Check on DISREAD. This error is posted when a parity error is detected on the DIS Channel during a read operation.

IBM Internal Use Only

DIS errors 10 - 16 (HEX) don't hang the machine on level 3.

- XX10 (HEX) - Timeout during Select operation. This error is posted when an attempt is made to address a macrofunction card and the 'RETURN' tag fails to go active within 175 microseconds.
- XX11 (HEX) - Timeout during data transfer. This error is posted when an attempt is made to transfer data to/from a macrofunction card and the 'RETURN' tag fails to go active within 175 microseconds.
- XX12 (HEX) - Timeout on data transfer to BIC. This error is posted when an attempt was made to transfer data to/from a BIC and the 'RETURN' tag failed to go active within 7 milliseconds. This is a hardware timeout on interrupt level 3.

IBM Internal Use Only

GLOSSARY

APU: Arithmetic Processing Unit-- Provides floating point and trigonometric function support. This device is located on the DSC I/O channel card in the 24 pin socket provided.

BIC: Block Interface Card-- A Distributed Interface System (DIS) component. An interface card located between the DSS microprocessor controller and the sensor I/O. Complete description is available in the DSS User Guide Vol. II.

DIS: Distributed Interface System-- A Distributed Sensor Subsystem (DSS) component. Is comprised of a complete family of "manufacturing hardened" sensor I/O using a "common bus" architecture.

DSC: Distributed System Controller-- A Distributed Sensor Subsystem (DSS) component. The microprocessor controller used for driving DSS family sensor based I/O.

DSS: Distributed Sensor Subsystem-- A manufacturing control system architecture made up of a microprocessor controller and a complete family of sensor I/O used for process control applications.

External Register: A specialized storage location, 8 bits in length, used for system input/output (I/O) operations.

Hazel: See "DSC".

Local Storage Register: General Purpose Register. A specialized storage location, 8 bits in length, used for arithmetic, memory address, and data operations.

Nucleus: Operating System-- Provides language support and all user functions for a computer system.

Partition: Memory sector-- A 32k-Word sector of Random Access Memory in a DSC IV system.

PLEX: Program Level Exit-- The operation of leaving the current system priority interrupt level routine.

Register Chapter/Block: An addressable local storage register group. Each Chapter/Block is comprised of 64 registers organized in groups(pages) of 8 registers. The Chapter/Block is selected via contents of the "L" register (see Hazel IV hardware documentation). The Hazel IV machine is configured with 8 Register Chapters/Blocks, selected (0-7) making a total of 512 local storage registers.

Register Page: An addressable local storage register group. Each "page" is comprised of 8 local storage registers, selected by the contents of the Local Storage Page Register (LSPR) (see Hazel hardware documentation).

SPI: Set Program Interrupt-- The operation of forcing a system priority interrupt within the current program routine.

Heading ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Heading References</u>
dump	DUMP2	B-1	Memory Dump to S/1 Diskette

Figure ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Figure References</u>
dmp	DUMP2	B-1	8:

Footnote ID's

<u>id</u>	<u>File</u>	<u>Page</u>	<u>Footnote References</u>
didoout	DIDO	40	1: 40
f1	DUMP2	B-2	2: B-2
f2	DUMP2	B-2	3: B-2, B-2
f3	DUMP2	B-3	4: B-3
vpgm	BURN	D-3	5: D-3, D-4
rprrm	BURN	D-4	6: D-4, D-5
vrđ	BURN	D-4	7: D-4
anpt	BURN	D-5	8: D-5
enuc	BURN	D-5	9: D-5

Imbed Trace

Page i	PREFAC
Page viii	INTRO
Page 1	SYSINIT
Page 4	BRNGUP3
Page 8	BRNGUP4
Page 17	CSSAPBU
Page 18	DISKAPBU
Page 20	DISINIT
Page 21	SUPUT
Page 34	DIAGPGM
Page 35	ADTST
Page 37	APUT
Page 38	DATST
Page 39	DIDO
Page 41	DCOMM
Page 44	EDSJ
Page 46	GASPJ
Page 48	MFTJ
Page 51	ROCDG
Page 54	RS232
Page 56	DUT
Page A-6	DSCDMP
Page B-1	DUMP2
Page B-8	DEBUG
Page C-1	TSTKEY
Page C-15	DB3101
Page C-30	BURN
Page D-8	COMMERR
Page E-2	DISERR
Page F-2	GLOSSARY